https://doi.org/10.20965/jaciii.2019.p0705

Paper:

Scalable Change Detection Using Place-Specific Compressive Change Classifiers

Kanji Tanaka

University of Fukui
3-9-1 Bunkyo, Fukui, Fukui 910-8507, Japan
E-mail: tnkknj@u-fukui.ac.jp

[Received January 20, 2018; accepted February 1, 2019]

With recent progress in large-scale map maintenance and long-term map learning, the task of change detection on a large-scale map from a visual image captured by a mobile robot has become a problem of increasing criticality. In this paper, we present an efficient approach of change-classifier-learning, more specifically, in the proposed approach, a collection of place-specific change classifiers is employed. Our approach requires the memorization of only training examples (rather than the classifier itself), which can be further compressed in the form of bag-of-words (BoW). Furthermore, through the proposed approach the most recent map can be incorporated into the classifiers by straightforwardly adding or deleting a few training examples that correspond to these classifiers. The proposed algorithm is applied and evaluated on a practical long-term cross-season change detection system that consists of a large number of place-specific object-level change classifiers.

Keywords: place-specific change classifiers, scalable change detection, zero-shot learning, bag-of-words scene model

1. Introduction

In this study, we consider the problem of scalable change detection using a novel, compact representation of an environment map (**Fig. 1**). With the recent progress in large-scale map maintenance [2] and long-term map learning [3], the task of change detection on a large-scale map using a visual image captured by a mobile robot, has become a problem of increasing criticality [4]. We aim to enhance the scalability of map compactness from a novel perspective, while maintaining the effectiveness of the change detection system.

In general, change detection has the goal of detecting changes between the view image of a robot and a previously constructed map. A major challenge in change detection is to effectively address appearance variations in the changed objects. As the variations are inherently place-specific [5] and attributed to various factors (e.g., objects, viewpoint, background, illumination conditions, and occlusions), it is challenging to obtain a general-purpose change model. Thus far, the most fundamental scheme reported for addressing this challenge is to directly compare each view image against the corresponding reference (i.e., mapped) image using handcrafted features [6] or deep-learning techniques [7]. However, this fundamental scheme requires explicit memorization of every possible mapped image and exhaustive many-to-many image comparisons, which severely limits the scalability in time and space.

Our approach is inspired by the recent success in the community of change-classifier-learning [8–11]. Instead of memorizing the large collection of mapped images, the

Fig. 1. Scalable change detection for long-term map maintenance. (a) Experimental environment. The trajectories of the four datasets "2012/01/22," "2012/03/31," "2012/08/04," and "2012/11/17" used in our experiments are visualized in colored curves, respectively; they are overlaid on the bird's-eye-view image obtained from the North Campus long-term (NCLT) dataset [1]. (b) Examples of changes. For each panel, the left and right images are a query image and its corresponding reference (i.e., mapped) image, respectively.

change-classifier-learning approach simply learns an essential change classifier that is often significantly compact and nonetheless exhibits a generalization capability. In [8], Gueguen et al. presented a change detection method for overhead imagery they trained a support vector machine (SVM) with linear kernel with L1-regularization and L2-loss as a semi-supervised scene-specific change classifier from manually labeled positive examples (i.e., changes) and a plethora of available negative examples (i.e., no-changes). Moreover, it is preferable to use only a single change classifier; however, it is challenging for a single classifier to capture the place-specific variations in changes [5] or to flexibly incorporate the most recent local changes in the map [12]. Therefore, we opted to use multiple place-specific change classifiers, each of which was responsible for each place-specific region; the place-specific change classifiers could be flexibly and locally updated, added, or deleted at a marginal fixed cost. Each classifier receives local feature-descriptors (e.g., oriented FAST and rotated BRIEF (ORB) descriptors) as input and evaluated their likelihood of change. However, it was not straightforward to apply the typical framework of change-classifier-learning to our application domain, i.e., autonomous robotics, in which examples with labels (i.e., change/no-change) are not provided. Instead, the mapper-robot itself should collect examples in an unsupervised manner.

In this study, we addressed the aforementioned issue from the viewpoint of zero-shot learning (ZSL) [13]. ZSL is a domain-adaptation technique in the field of machine learning, it was originally proposed as a strategy to obtain classifiers for arbitrary, novel categories when labeled examples are not available. In particular, we were inspired by mining-based ZSL [14], where the training examples for novel unseen categories (in the target domain) are collected by mining an external knowledge base (EKB) such as a search engine. To realize our ZSL-based change-classification framework, it was necessary to address the following questions: (1) How does one train classifiers? (2) How does one represent examples? (3) How does one prepare an EKB?

The key concepts required for answering these questions are the following:

1. We trained classifiers using the most recent examples when necessary to incorporate the latest changes in the map;

2. We used bag-of-words (BoW) as a compact and discriminative representation of examples;

3. We employ a visual word vocabulary as the EKB,

which were inspired by the use of BoW as a compressed form of place-specific classifier in our previous study [15]. The aforementioned concepts exhibited several advantages: (1) Only training examples needed to be maintained, rather than the classifier themselves because typical classifier-learning algorithms such as SVM [16] and the nearest neighbor (NN) [17] allow us to reproduce a classifier, given the same training examples [15]. (2) We were able to incorporate the latest changes into the classifiers by simply adding or deleting a few training examples that corresponded to these classifiers, meanwhile the training-time overhead per place could remain reasonably low with efficient training algorithms such as NN and SVM. (3) We were able to represent each training example in a significantly compact form of a visual word [15], which implied that the training example was approximated by its NN exemplar feature from a visual word vocabulary. (4) We were able to share the vocabulary with other task related scenarios including BoW-based self-localization, simultaneous localization and mapping (SLAM), lifelong learning, and open-set recognition, in which the vocabulary is typically designed as a rich, continuously growing, and domain-adaptive knowledge base. Through experiments, the proposed framework was evaluated in a cross-season change detection setting using the publicly available NCLT dataset [1].

1.1. Relation to Other Works

The majority of the present state-of-the-art change detection systems are based on image differencing or pairwise image comparison [5]. In [18], a scene alignment method for image differencing was proposed based on ground surface reconstruction, texture projection, image rendering, and registration refinement. In [4], a deep deconvolutional network for pixel-wise change detection was trained and used for comparing query and reference image patches.

In certain recent studies use change-classifier-learning was used to realize more efficient and compact change detection [8–11]. Our algorithm was inspired by these classifier-learning approaches and advances a step further from the perspective of unsupervised ZSL. The work of Gueguen and Hamid [8] can be considered as one of the works that is most relevant to our study. In their work, they addressed change detection for overhead imagery. A BoW model with tree-of-shape features was employed to achieve more effective accuracy-efficiency tradeoff. Based on these features, linear canonical correlation analysis was employed to learn a subspace to encode the notion of change between images. To reduce the cost of label acquisition by human photo-interpreters, a semi-supervised SVM framework was introduced. In contrast, we will address unsupervised settings, where the examples need to be labeled automatically by the robot itself rather than by human labelers.

Our study is focused on a monocular camera as the sole input device. This is a significantly challenging setting compared with other change detection settings, in which a richer information source is assumed, including a 3D model [19], a stereo or image sequence [6], 3D data [20], multi-spectral overhead imagery [21], and an object model [22].

In the area of field robotics, substantial work exists on change-detection systems for patrolling [23], agri-

culture [24], tunnel inspection [25], and damage detection [8]. We consider that our extension would also contribute to these applications from the novel perspective of map compactness.

This work is a part of our study on compact map model for scalable change detection and long-term map maintenance [15, 26–29]. In our previous studies, two scenarios were considered. One was the "global localization" scenario [28], wherein the change detection system was required to function under global viewpoint uncertainty. As this scenario required access to the entire large-size maps in memory, individual images were used in their compressed form of BoW [26]. In contrast, the current study is focused on the alternative "pose-tracking" scenario, in which the system can assume that the viewpoint of the robot is tracked over time. A similar setting was addressed in our recent study [27]; however the most important difference of the previous study to the present one is that the classifiers were not compressed. As the pose-tracking scenario requires access to only a marginal portion of the map that corresponds to the surroundings of the robot, mapped images may be used in a less-compact form of the decompressed classifiers. Nevertheless, it is necessary to maintain the remaining classifiers compressed. Hence, the efficiency of compression/decompression is an additional critical topic, which will be discussed in the present paper as well.

2. Approach

Our aim is to enhance the compactness of map representation for scalable change detection. As mentioned earlier in the manuscript, this rules out typical image-differencing approaches that require memorization of a number of high-resolution mapped images that are proportional to the map size. Therefore, we decided to use the change-classifier approach. More specifically, we employed multiple place-specific change classifiers to capture the place-specific nature of changes and to enable a flexible and local update in the change model. The main concept was to represent each classifier by their training examples and to compress each training example to a visual word; this enabled the development of a significantly compact and domain-adaptive change model.

Based on the above consideration, we use a set of place-specific change classifiers and represent each by a BoW. According to [30], a BoW is represented as an unordered collection of visual words, i.e., local visual features vector quantized by a codebook of exemplar features referred to as vocabulary. In our approach, an i-th classifier was represented by positive (i.e., change) and negative (i.e., no-change) sets of visual words, namely S_i^+ and S_i^-, respectively. Each visual word $w \in S_i^{\{+,-\}}$ is represented in the form:

$$w = \langle w^a, w^r \rangle. \quad \ldots \quad (1)$$

where w^a is a B-bit code referred to as appearance word and is an identifier for its NN exemplar-feature in the vocabulary. We employed an ORB feature descriptor [31] as a local-feature-descriptor because it is a rapid binary keypoint descriptor that has been applied to numerous real-time computer-vision and robotic-mapping problems [32]. The number of ORB descriptors per image was set to 2000. We denote the appearance-word vocabulary as V (i.e., $B = \log_2 |V|$). w^r is a pose word and represents the spatial location of the feature keypoint descriptor with respect to the object region; it incurs a small constant space cost $B' = \log_2 |A|$ (bits) where $|A|$ is the area (pixels) of the object region. Consequently, the total space-cost (bits) of our BoW-based model may be approximately calculated using the following expression:

$$C = \sum_{x \in X} \left[\sum_{r \in R(x)} \left[\sum_{w \in W(r)} \log_2 |V| + B' \right] \right], \quad \ldots \quad (2)$$

where X is the set of place regions, $R(x)$ is the object regions (described in Section 2.1) in the x-th place, and $W(r)$ are the visual words belonging to the object r.

Apparently, it is necessary to minimize the sizes of X, $R(x)$, $W(r)$, and V without compromising the accuracy of the change-classification. In general, their joint minimization is intractable, in this study, we will straightforwardly consider each minimization problem separately. The minimization of $R(x)$, $W(r)$, and V is the task of change-aware object proposal, feature extraction, and vocabulary design. We have addressed the issue of vocabulary design and the use of BoW as a compressed form of classifiers in an alternative context of visual self-localization in [15]. The minimization of X is referred to as unsupervised place-definition and workspace-partitioning discovery. This problem has been addressed in our previous studies in the context of place recognition [33] and change recognition [27]. In this study, we straightforwardly partition the entire sequence of mapped images in the workspace into equal-length subsequences, each of which corresponds to each place class.

Our algorithm consists of two distinct phases: training and classification. The training procedure consists of the following steps: (t1) Train classifiers. (t2) Represent classifiers by their positive/negative examples. (t3) Approximate the examples by BoWs. (t4) Memorize the examples in the form of BoW. The classification procedure consists of the following steps: (c1) Lookup the vocabulary to determine examples that correspond to the specified reference image. (c2) Reproduce classifiers from the examples. (c3) Classify query features using the reproduced classifiers. (c4) Delete classifiers. It should be noted that steps (c2) and (c4) function as the decompression and compression of classifiers, respectively (**Fig. 2**). The time overhead of the compression task is relatively negligible, whereas that of the decompression task is marginal albeit noticeable. Therefore, it is necessary to adequately perform task scheduling of the decompression task. In this study, we follow a straightforward strategy: when the robot approaches or enters a new submap region, the classifiers of its surrounding submaps are decompressed.

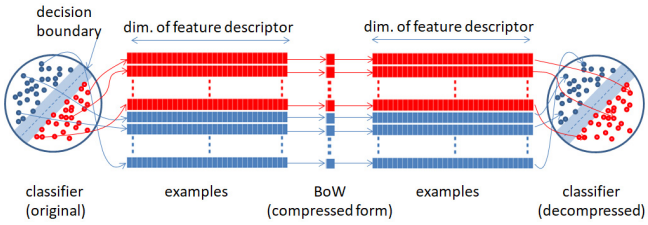


Fig. 2. Pipeline of the classifier-compression/-decompression task. Top: positive examples. Bottom: negative examples.

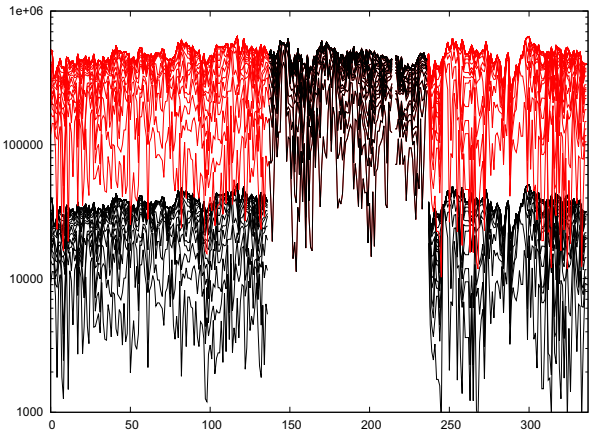


Fig. 3. Effect of map compression. Vertical axis: Total space cost (bits) for all the place-specific change classifiers. Horizontal axis: Frame ID. Black and colored curves exhibit space costs for compressed and non-compressed (or decompressed) maps. Each k-th curve corresponds to the k-th object cluster; the intervals between the $(k-1)$-th and k-th curves are equal to the space cost of the k-th object cluster.

Figure 3 demonstrates our strategy. In the figure, the frame IDs ranging from 137 to 237 correspond to the classifiers for the surroundings of the robot. As illustrated, the space cost of the compressed classifiers is significantly lower than that of non-compressed or decompressed classifiers, which demonstrates the effectiveness of our approach.

2.1. Classifier-Learning

The place-specific change model aims to detect changes at the object level. It consists of a set of object-level classifiers, each of which aims to learn the appearance of a known object (i.e., no-change object) in the reference image of a specific place and then, to classify an unseen query feature as either change or no-change with respect to the learned object. To this end, object segmentation both at training and at classification tasks significantly influences the performance of the object-level change detection.

The training phase begins with the extraction of a collection of object proposals from the specified reference image (**Fig. 4(b)** left). To achieve this, we use the binarized normed gradients (BING) object-proposal algorithm [34] because it is highly efficient to provide object proposals with category-independent image windows. Subsequently, the proposed object-regions are further grouped into clusters of spatially-near regions, termed object clusters (**Fig. 4(b)** right). We use a criterion in this clustering where any two object regions belong to the same cluster if their bounding boxes overlap one another. Next, an additional "background" object cluster, which treats the entire image region as the object region, is added. Then, a change classifier is trained for each object cluster. We will implement and evaluate two types of classifiers: namely the NN, or SVM, and four types of SVM kernels, i.e., the linear, the sigmoid, the polynomial, or the radial basis function (RBF). The aforementioned methods of object proposal and classifier-learning are detailed below.

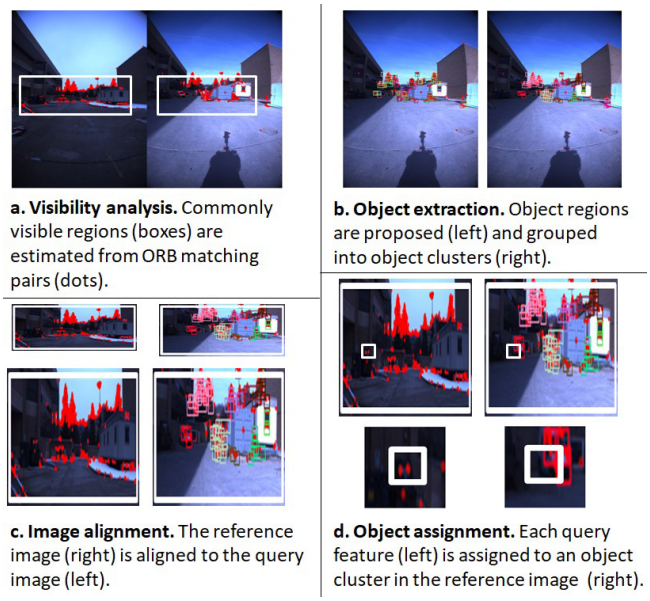


Fig. 4. Overview of scene recognition.

In general, the BING algorithm produces a large number of object proposals (e.g., 2×10^3 proposals per image). Typically, the proposals contain numerous false positives. Evaluating all the proposals is computationally intractable. Therefore, we decide to select a marginal portion of the object proposals. From the viewpoint of effectiveness of our BoW model, we evaluated the spatial density of visual words N/A, where A and N are the area of the bounding box of the object proposal and the number of visual words inside the bounding box, respectively. Then, we selected 400 object proposals with the highest density. Subsequently, near-duplicate object regions that overlapped significantly with object regions with higher density were eliminated. We considered two object regions i and j to be in the near-duplicate condition if their overlap ratio $A_{ij}/\min(A_i, A_j)$ exceeds a threshold of 0.5; here, A_i, A_j, and A_{ij} are the areas (pixels) of the objects i and j and their overlap region, respectively.

Training examples for change classifiers were collected

via mining-based ZSL. The change classifier addressed the one-sided classification problem, where only negative examples (i.e., no-change) were provided during the training stage. To obtain a sufficiently large number of positive examples, we considered the ZSL task which refer to as change mining [27], the aim of which is to mine features of potential change objects from a large EKB feature collection. In this study, we considered three strategies for change mining: “uniform,” “farthest,” and “nearest.” The “uniform” strategy uniformly samples positive examples from the EKB, while the “farthest” and “nearest” strategies sample positive examples that are the farthest and nearest in Hamming distance from the specified negative examples, respectively. For all the aforementioned strategies, each example in the EKB that was nearer than 10 bit from its NN negative was considered inappropriate as a positive example and was eliminated from the candidates of positive examples prior to the change mining. To enable efficient training, the number of training examples was truncated to a maximum number of 400.

2.2. Change Classification

The change classification process aims to rank local feature-descriptors in query images in the order of the likelihood of changes. It consists of two distinct steps: (1) image registration and (2) change ranking. Both steps are detailed below.

The image registration step, whereas aims to align query and reference images into the same coordinate frame (**Fig. 4(c)**), is a necessary pre-processing step for a substantial majority of change detection tasks [5]. It should be noted that image alignment from monocular image-pairs is significantly ill-posed, particularly when we are provided only BoW representations of the images. We tested three strategies for feature matching – ORB keypoint matching with and without post-verifications using random sample consensus (RANSAC) [35] and using vector field consensus (VFC) [36] – and observed that the aforementioned post-verification strategies (i.e., RANSAC and VFC) were not adequately stable in our scenario of highly-complex scenes. Therefore, we used ORB keypoint matching as a method for image alignment.

In this work, we followed a fundamental procedure for image registration [18], which assumes a linear transformation from reference- to query-image coordinate (**Fig. 4(c)**). However, an important difference was that we were given the BoW representation instead of raw feature descriptors. It should be noted that the transformation algorithm requires pairs of feature keypoints matched between query and reference images. To filter out outlier matches to the maximum, an image region commonly visible in both images was estimated in the form of a bounding box (**Fig. 4(a)**). More formally, keypoints of matched visual words M were sorted in ascending order of x- or y-coordinates, we define $\lfloor \delta |M| \rfloor$-th and $\lceil (1-\delta)|M| \rceil$-th elements ($\delta = 0.1$) in the sorted lists as the x- and y-locations of the boundary of the visible regions.

Two types of additional spatial cues were obtained as a byproduct of the image registration. The first one was global spatial-information for visibility analysis (**Fig. 4(a)**), where the local features outside the commonly visible region were not considered as change-object candidates. The second was local spatial-information by which each object region (i.e., bounding box) in the reference image was first transformed to the coordinate of the query image (**Fig. 4(c)**); following this, the local features of the query image outside the transformed bounding box (**Fig. 4(d)**) were not considered as matching candidates of the specific object in the reference image. Considering the transformation errors, we expanded the transformed bounding box by ΔL (pixel), where margin ΔL was set to 10% of the image width.

The change-ranking step aims to rank query features by the likelihood of change, given the learned place-specific object-level change classifiers. First, each query feature was assigned to the object cluster that is spatially-near in the corresponding reference image (**Fig. 4(d)**). We assigned each query feature to an object cluster if its keypoint is located within one of the bounding box of the object cluster. Then, each query feature was used as input to the assigned change classifier of the object cluster, the classifier would output the probability p_r of the query feature not originating from the learned object cluster r. We employed Platt scaling [37] with five-fold cross-validation to convert an output value of the SVM classifier to the probability estimate or use a Gaussian $[1-\exp(-d^2/\sigma_d^2)]$ to convert an output distance of the NN classifier d into a probability value. Given the output probability value p_c for each object cluster c, the probability of the query feature of interest changing was computed as:

$$p = \min_c p_c. \quad (3)$$

To render the set of top-ranked features more diverse, we introduce the idea of non-maximal suppression and penalize features that are considered similar to a higher-ranked feature. In our view, two features are similar if they belong to the same object cluster. Based on this concept, we extracted object clusters from a specified query image and grouped query features into clusters in a similar manner as in the reference images; then, features in each cluster were ranked in descending order of the probability p. Next, the entire set of query features was re-ranked in ascending order of augmented score $r + (1-p)$, where $r(\in [1,C])$ is the abovementioned intra-cluster rank, C is the size of the largest cluster, and $p(\in [0,1])$ is the original probability-estimate.

It should be noted that not all query features always belong to object clusters; certain query features may exist that are isolated and do not belong to any object cluster. We empirically determined that such isolated features are in general less-reliable. Therefore, we assigned them a worst intra-cluster rank $r = C+1$.

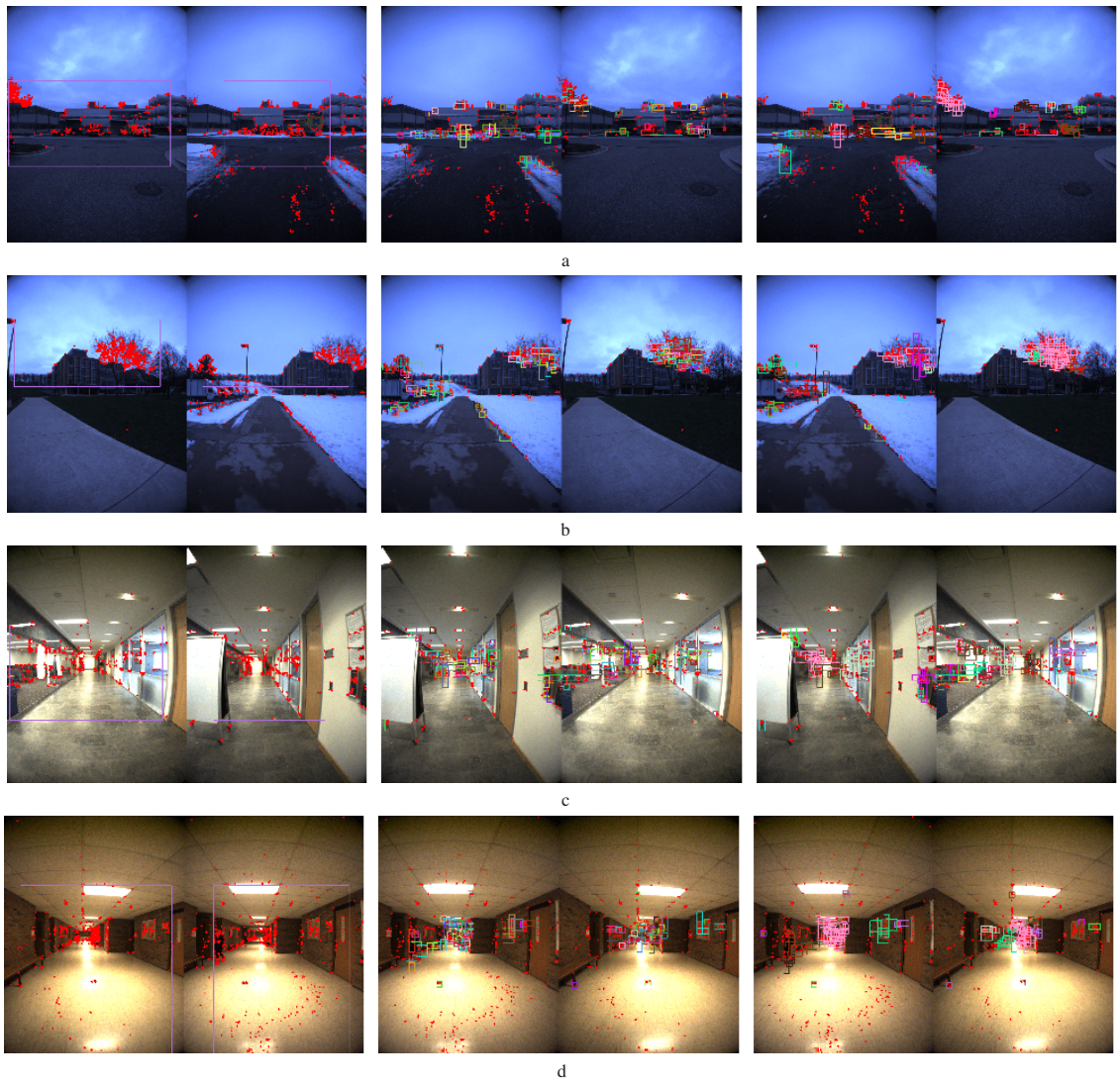


Fig. 5. Examples of scene recognition. Left panel: ORB keypoints (dots) and commonly visible regions estimated between query and reference images (boxes). Middle and right panels: Object regions and object clusters extracted from both images, using different colors for different regions and clusters.

3. Experiments

We evaluated various change detection strategies using the NCLT dataset [1]. The NCLT dataset is a long-term autonomy dataset for robotics research collected in the North Campus of the University of Michigan. The dataset consists of omnidirectional imagery, 3D lidar, planar lidar, GPS, and odometry: we used the monocular images from the front-directed camera ("camera #5") for our change detection tasks. **Figs. 5** and **6** illustrate certain examples of scene recognition.

During the travel of the vehicle through both indoor and outdoor environments (**Fig. 1(a)**), it encounters various types of changes, which originate from the movement of individuals, the parking of cars, the furniture, the building construction, the opening/closing of doors, and the placement/removal of posters (**Fig. 1(b)**). Moreover, nuisance changes exist that originate from illumination alterations, viewpoint-dependent changes of the appearance and occlusions of objects, weather variations, falling leaves and snow. A critical and significant challenge in a substantial number of change detection tasks is to distinguish changes of interest from nuisances. This renders our change detection task significantly more demanding.

We use four datasets, namely "2012/1/22," "2012/3/31," "2012/8/4," and "2012/11/7" that correspond to four sessions of vehicle navigation. These datasets consisted of 5095, 3994, 4877, and 5118 images with a size of 1232×1616. We manually created 7571 pairs of corresponding query and reference images, which corresponded to "revisiting" or "loop-closing" situations [28], using the global viewpoint information. Although 548 of the pairs were change image-pairs, 6714 pairs were no-change image-pairs; the remaining 309 pairs were not independent of the 548 change pairs and were not used in the experiments. We categorized small changes (e.g., 10×10 pixels) which, typically, originated from distant objects, into no-change because

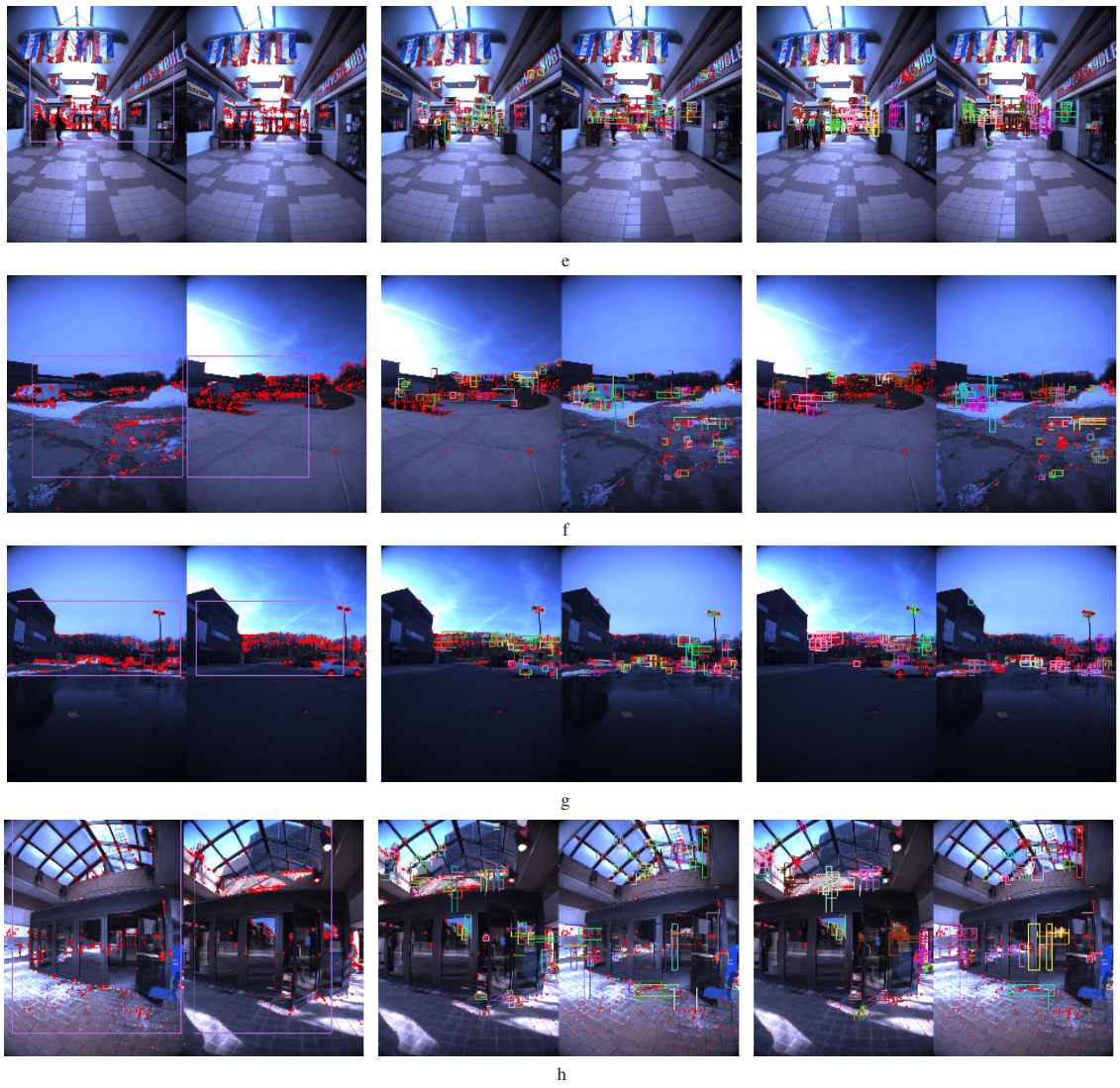


Fig. 6. Examples of scene recognition. Left panel: ORB keypoints (dots) and commonly visible regions estimated between query and reference images (boxes). Middle and right panels: Object regions and object clusters extracted from both images, using different colors for different regions and clusters.

it was challenging to detect such small changes through a visual change detection algorithm. As a result, change detection algorithms objects were likely to yield a false-positive detection for such small objects. We used a specific combinations of query and reference image sets: (query, reference) = (2012/1/22, 2012/3/31), (2012/3/31, 2012/8/4), (2012/8/4, 2012/11/7), or (2012/11/7, 2012/1/22). The size of the vocabulary was 1 M (i.e., 20-bit visual words) by default, and its exemplar features were randomly sampled from the visual features of the reference set. The change objects in the 548 change image pairs were manually annotated in the form of bounding boxes.

The average overhead time for compression and decompression per classifier was 0.3 s and the overhead time per place was 3.0 s using a non-optimized implementation of SVM from SVM light [38] on a laptop (Intel Core i5-4200 2.5 GHz). This implies that the compression/decompression can function in real-time when the robot moves at a speed of 1 m/s; the workspace was divided into places with a length of 3 m. The total travel distance was 22,181 m.

We considered a straightforward change detection task scenario: given a collection of 100 query images acquired by a robot, the changed image and the locations of the change features within that image should be identified. The size of an image collection was set to 100, and it consisted of one changed image and 99 random no-change images. We created 548 collections based on the aforementioned 548 changed image pairs. For each collection, 99 no-change images were randomly sampled from the 6714 no-change pairs; the collections were commonly used as dataset by each change detection algorithm.

We considered various strategies of change mining, classifiers, and image registration. For the change mining, we implemented three methods: "uniform," "farthest," and "nearest," which have been described in Section 2.1. For the classifier, we implemented the SVM

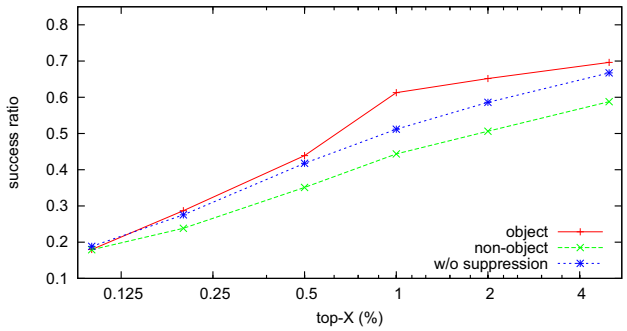


Fig. 7. Change detection performance.

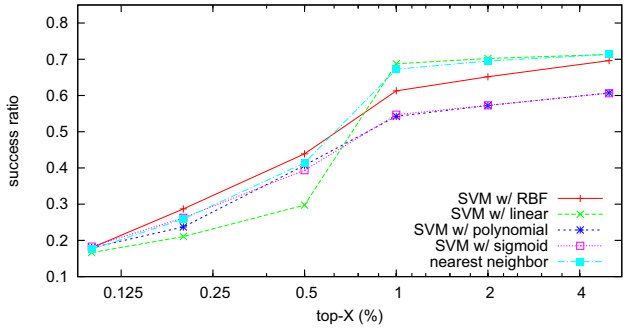


Fig. 8. Influence of classifiers and kernels.

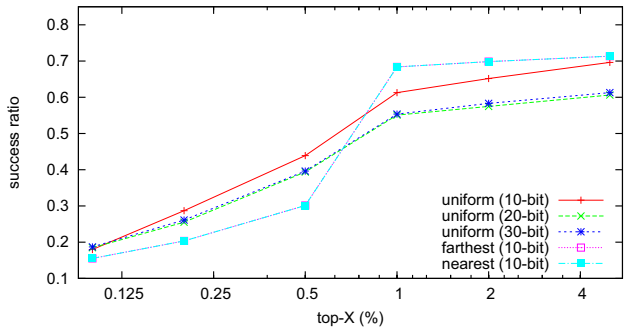


Fig. 9. Influence of change-mining strategies.

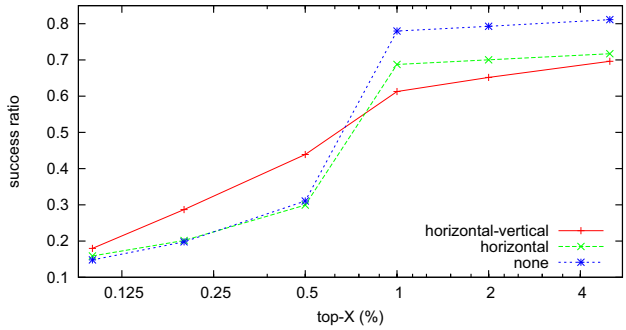


Fig. 10. Influence of visibility-analysis strategies.

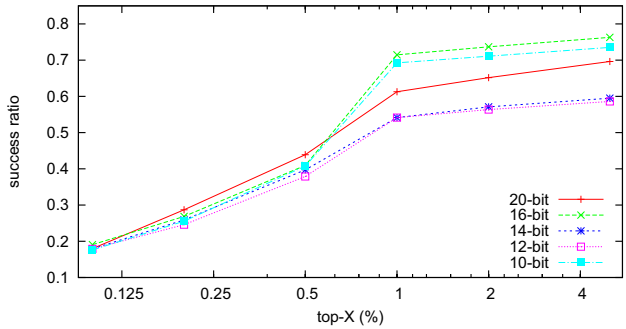


Fig. 11. Influence of vocabulary size.

classifier with four kernels – linear, sigmoid, polynomial, and RBF – and the NN classifier with the Hamming distance metric. For the image registration, we aimed to verify the efficacy of the visibility analysis strategy presented in Section 2.2; we implemented the common visible region in the form of a bounding box ("horizontal-vertical") as well as two additional methods for visibility analysis. One of these two was a method in which the y-direction boundaries are omitted; this omission was motivated by the observation that y-direction spatial information is typically, unreliable ("horizontal"). The other was a method that entirely omitted the bounding-box information, which corresponded to not using the visibility analysis strategy ("none"). By default, we used SVM with an RBF kernel as the classifier, a uniform sampling with a 10-bit distance threshold as the change mining strategy, and the 20-bit vocabulary.

Figures 7–11 illustrate the performance results. We evaluated the various methods using the independent 548 image collections. Each image collection consisted of 100 pairings of query and reference images; each query image contained 2000 features to score. We assessed whether the change detection task on an image collection is successful or not and compute the success ratio for all the image collections. For the assessment, all $100 \times 2{,}000 = 200{,}000$ features were sorted in ascending order of the augmented score (as described in Section 2.2). If a ground-truth change feature would be ranked within the top-$X[\%]$ with respect to the sorted list, the change detection would be considered as a success; otherwise, it would be considered as a failure. We evaluated various X values: 0.1%, 0.25%, 0.5%, 1%, 2.5%, and 5%, which respectively corresponded to the top 200, 500, 1000, 2000, 5000, and 10000 ranked query features, respectively.

Figure 7 illustrates the results of the evaluation of the fundamental effects of the proposed strategies – object-level change detection (Section 2.1) and non-maximal suppression (Section 2.2). We developed a comparison method referred to as "non-object," which employs a single image-level classifier rather than the object-level classifier. The image-level classifier was implemented as a single object-level classifier that treated the entire image as the object region. In addition, we developed another comparison method referred to as "non-suppression," which did not employ the non-maximal suppression technique. In the figure, "object" indicates the proposed method (classifier: SVM w/ RBF kernel, vocabulary size of 20 bit); "non-object" and "non-suppression" indicate the two comparison methods. It may be observed that the proposed "object" method evidently outperforms the remaining; it may verified that the object-level change classifiers are substantially more powerful than the comparison feature-level methods.

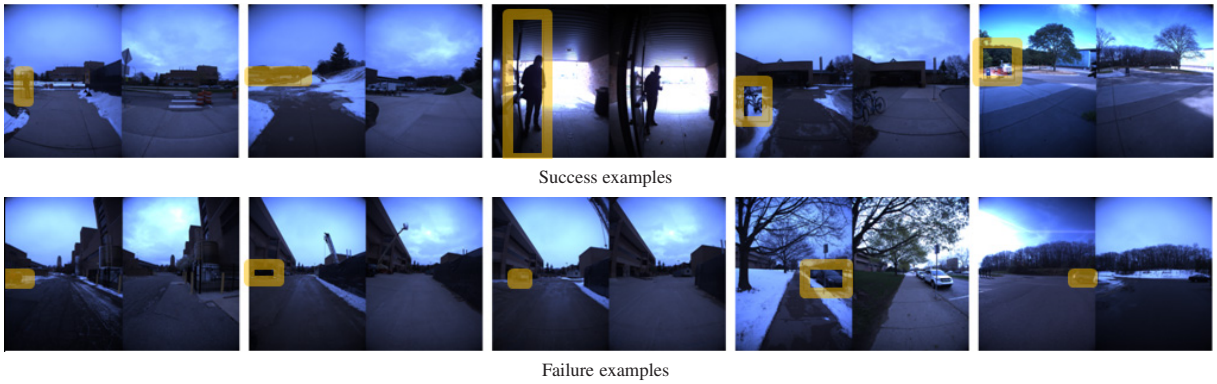


Fig. 12. Change detection examples.

Figure 8 illustrates the comparison between classifiers and kernels. We evaluated different classifiers and kernels (NN, SVM with linear, sigmoid, polynomial, and RBF kernels). It may be observed that SVM with an RBF kernel outperforms the other classifiers and kernels on the left side of the graph ($X < 1.0$). However, it is inferior to the SVM with linear kernel and NN on the right side of the graph ($X \geq 1.0$).

Figure 9 shows the comparison of different change mining strategies. We evaluated different change mining strategies (uniform with a 10-bit, 20-bit, and 30-bit distance threshold, farthest, and nearest). It may be seen that a uniform sampling with a 10-bit distance threshold outperforms the other methods. This may be attributed to the fact that the uniform sampling strategy tends to produce a more diverse set of positive examples than other strategies; thus the resulting classifier exhibits adequate generalization capability.

Figure 10 illustrates the evaluation results of the visibility analysis strategies. We evaluated different visibility analysis strategies (horizontal-vertical, horizontal, and none). It may be observed that the horizontal-vertical strategy outperforms the remaining two strategies when parameter X is sufficiently marginal.

Figure 11 illustrates the results of various vocabulary sizes, namely, 20, 16, 14, 12, and 10 bit. It may be observed that the change detection performance is the highest for the largest vocabulary (i.e., 20 bit). It is noteworthy that the 10-bit-vocabulary yields a reasonably high performance, while its space cost for the visual words and the vocabulary are 50% and 0.1%, respectively, compared with those of the case of the 20-bit vocabulary.

Figure 12 illustrates examples of successful and failed change detection. Here, we used the aforementioned default settings. As illustrated, changes originating from moving objects such as cars and pedestrians are generally easy to detect owing to the fact that their visual appearances are significantly discriminative from other background objects such as roads and trees. However, the detection of moving objects becomes a substantially challenging task when a similar moving object appears in the corresponding reference image (**Fig. 12** failure examples). Although spatial information (e.g., location and size) of such similar objects is, in general, dissimilar to that of the query object, these objects are typically accepted as matching candidates of the query object because the search region is expanded to overcome errors in coordinate transformation (as explained in Section 2.2). Other challenging cases include change objects such as boxes and tables (**Fig. 12** failure examples) whose visual appearance is significantly similar to background objects such as floors and walls.

4. Conclusions

We presented a change detection framework that realized map compactness while maintaining detection efficiency. Rather than memorizing pre-trained classifiers, our ZSL-based approach only memorized the compact indices to their training examples that had been mined from an EKB. The experimental results of place-specific object-level change classifiers demonstrated a high potential. The proposed algorithm was efficient and very simple to implement. Therefore, it is convenient to integrate this algorithm into existing frameworks of change detection (e.g., change detection in 3D, stereo images) to enhance their compactness.

Acknowledgements

Our work has been supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) 26330297, and for Scientific Research (C) 17K00361.

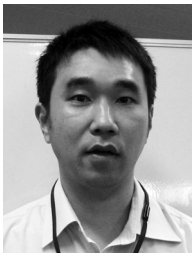
References:

[1] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," Int. J. of Robotics Research, Vol.35, No.9, pp. 1023-1035, 2016.

[2] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3D map maintenance in dynamic environments," Proc. of 2014 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3712-3719, 2014.

[3] M. Fehr, M. Dymczyk, S. Lynen, and R. Siegwart, "Reshaping our model of the world over time," Proc. of 2016 IEEE Int. Conf. on Robotics and Automation (ICRA) pp. 2449-2455, 2016.

[4] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-View Change Detection with Deconvolutional Networks," Autonomous Robots, Vol.42, No.7, pp. 1301-1322, 2018.
[5] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," IEEE Trans. image processing, Vol.14, No.3, pp. 294-307, 2005.
[6] J. Košecka, "Detecting changes in images of street scenes," Proc. of the 11th Asian Conf. on Computer Vision (ACCV) – Volume Part IV, pp. 590-601, 2012.
[7] S. Stent, R. Gherardi, B. Stenger, and R. Cipolla, "Detecting Change for Multi-View, Long-Term Surface Inspection," Proc of the 26th British Machine Vision Conf. (BMVC), pp. 127.1-127.12, 2015.
[8] L. Gueguen and R. Hamid, "Large-scale damage detection using satellite imagery," Proc. of 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1321-1328, 2015.
[9] A. A. Nielsen, "The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data," IEEE Trans. on Image Processing, Vol.16, No.2, pp. 463-478, 2007.
[10] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel approach to unsupervised change detection based on a semisupervised SVM and a similarity measure," IEEE Trans. on Geoscience and Remote Sensing, Vol.46, No.7, pp. 2070-2082, 2008.
[11] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," IEEE Trans. on Geoscience and Remote Sensing, Vol.38, No.3, pp. 1171-1182, 2000.
[12] B. Yamauchi and P. Langley, "Place learning in dynamic real-world environments," Proc. RoboLearn, Vol.96, pp. 123-129, 1996.
[13] E. Gavves, T. Mensink, T. Tommasi, C. G. M. Snoek, and T. Tuytelaars, "Active transfer learning with zero-shot priors: Reusing past datasets for future tasks," Proc. of 2015 IEEE Int. Conf. on Computer Vision (ICCV), pp. 2731-2739, 2015.
[14] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol.36, No.3, pp. 453-465, 2014.
[15] K. Tanaka, "Cross-season place recognition using NBNN scene descriptor," Proc. of 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 729-735, 2015.
[16] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, Vol.20, No.3, pp. 273-297, 1995.
[17] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. on Information Theory, Vol.13, No.1, pp. 21-27, 1967.
[18] D. W. J. M. van de Wouw, G. Dubbelman, and P. H. N. de With, "Hierarchical 2.5-D Scene Alignment for Change Detection With Large Viewpoint Differences," IEEE Robotics and Automation Letters, Vol.1, No.1, pp. 361-368, 2016.
[19] A. Taneja, L. Ballan, and M. Pollefeys, "Geometric change detection in urban environments using images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.37, No.11, pp. 2193-2206, 2015.
[20] J. P. Underwood, D. Gillsjö, T. Bailey, and V. Vlaskine, "Explicit 3D change detection using ray-tracing in spherical coordinates," Proc. of 2013 IEEE Int. Conf. on Robotics and Automation, pp. 4735-4741, 2013.
[21] W. Li, X. Li, Y. Wu, and Z. Hu, "A novel framework for urban change detection using VHR satellite images," Proc. of 18th Int. Conf. on Pattern Recognition (ICPR'06), Vol.2, pp. 312-315, 2006.
[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.39, No.6, pp. 1137-1149, 2017.
[23] H. Andreasson, M. Magnusson, and A. Lilienthal, "Has something changed here? Autonomous difference detection for security patrol robots," Proc. of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3429-3435, 2007.
[24] P. Ross, A. English, D. Ball, B. Upcroft, G. Wyeth, and P. Corke, "Novelty-based visual obstacle detection in agriculture," Proc. of 2014 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1699-1705, 2014.
[25] S. Stent, R. Gherardi, B. Stenger, K. Soga, and R. Cipolla, "An Image-Based System for Change Detection on Tunnel Linings," MVA 2013 IAPR Int. Conf. on Machine Vision Applications, pp. 359-362, 2013.
[26] T. Murase, K. Tanaka, and A. Takayama, "Change Detection with Global Viewpoint Localization," 2017 4th IAPR Asian Conf. on Pattern Recognition (ACPR), pp. 31-36, 2017.
[27] X. Fei and K. Tanaka, "Unsupervised Place Discovery for Place-Specific Change Classifier," Computing Research Repository (CoRR), 1706.02054, 2017.
[28] K. Tanaka, Y. Kimuro, N. Okada, and E. Kondo, "Global localization with detection of changes in non-stationary environments," Proc. of 2004 IEEE Int. Conf. on Robotics and Automation (ICRA), Vol.2, pp. 1487-1492, 2004.
[29] M. Ando, Y. Chokushi, K. Tanaka, and K. Yanagihara, "Leveraging image-based prior in cross-season place recognition," Proc. of 2015 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 5455-5461, 2015.
[30] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," Proc. of Int. workshop on multimedia information retrieval, pp. 197-206, 2007.
[31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 Int. Conf. on Computer Vision, pp. 2564-2571, 2011.
[32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE Tran. on Robotics, Vol.31, No.5, pp. 1147-1163, 2015.
[33] X. Fei, K. Tanaka, K. Inamoto, and G. Hao, "Unsupervised place discovery for visual place classification," 2017 15th IAPR Int. Conf. on Machine Vision Applications (MVA), pp. 109-112, 2017.
[34] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," Proc. of 2014 IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3286-3293, 2014.
[35] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: a universal framework for random sample consensus," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.35, No.8, pp. 2022-2038, 2013.
[36] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," IEEE Trans. on Image Processing, Vol.23, No.4, pp. 1706-1721, 2014.
[37] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," Proc. of ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, pp. 213-220, 2008.
[38] T. Joachims, "Making large-Scale (SVM) Learning Practical," B. Schölkopf, C. Burges, and A. Smola (Eds.), "Advances in Kernel Methods-Support Vector Learning," pp. 169-184, MIT Press, http://svmlight.joachims.org, 1999 [accessed April 11, 2016]



Name:
Tanaka Kanji

Affiliation:
University of Fukui

Address:
3-9-1 Bunkyo, Fukui, Fukui 910-8507, Japan

Brief Biographical History:
1997 Received M.S. degree, Graduate School of Information Science and Electrical Engineering, Kyushu University
2000 Received Ph.D. degree, Graduate School of Information Science and Electrical Engineering, Kyushu University
2000- Research Assistant, School of Systems Information Science, Future University Hakodate
2002- Research Associate, Department of Intelligent Machinery and Systems, Kyushu University
2008- Associate Professor, Graduate School of Engineering, University of Fukui

Main Works:
• Current research interests are intelligent robotics, machine learning and pattern recognition.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• The Robotics Society of Japan (RSJ)
• The Institute of Electronics, Information and Communication Engineers (IEICE)
• The Society of Instrument and Control Engineers (SICE)
• The Japan Society of Mechanical Engineers (JSME)