Interval Inspired Approach Based On Temporal Sequence Constraints To Place Recognition

Renata Neuland[1], Fernanda Rodrigues[1], Diego Pittol[1], Renan Maffei[1], Mariana Kolberg[1] and Edson Prestes[1]

***Abstract*— Visual place recognition is a fundamental task in many robotics applications. Recognizing if the robot is crossing through an already visited place is required to improve its localization and map estimation. However, this is a challenging task considering the dynamicity of the environment. We propose a novel approach to place recognition inspired by interval analysis theory. We model the world as a set of intervals of observations and search for matchings based on nearest neighbors and temporal constraints. We demonstrate the effectiveness of our method by testing it using two challenging datasets.**

I. INTRODUCTION

The visual place recognition problem, also called loop closure detection, consists in recognizing previously visited places based on visual sensing [1]. Despite the difficulty involved in this process, due to dynamicity of the environment over time, many robotics applications need to deal with it to improve robot pose estimation and/or the quality of the map that is being built. Besides, the place recognition strategy must be as accurate as possible while avoiding the generation of false positives, since few erroneous matches are usually enough to cause the degradation of the SLAM process.

The use of vision as main sensing modality has presented exciting results for learning and recognizing places [2]. Nevertheless, challenging aspects such as vehicles movements, lighting, and natural or human-made structural changes need to be taken into consideration. Figure 1 shows examples of challenging situations; each line presents two images from the same place taken at different instants during a day. We can see both lighting and point of view changes.

Any place recognition method must have a representation of the environment, which is compared to new incoming data to find matchings. To determine if the current incoming data is from a place previously included in its representation of the environment, and if so, which one [5].

Along the past years, several methods were proposed to deal with this problem. FAB-MAP [6] and SeqSLAM [2] are the most popular approaches due to the quality of the results they obtained in challenging situations. Which guaranteed to them the recognition of being milestones in the state-of-art.

We propose a novel method for visual place recognition under environmental condition changes using a monocular camera. Our approach is inspired from interval analysis theory [7]. Which lies on the idea of enclosing numbers in intervals. It searches the solution by using constraints of the problem of interest to reduce the set of candidate solutions. If a candidate satisfies the constraints, it is part of the solution. Our method uses intervals to represent regions of the real-world with the support of information collected by robot sensor. It creates constraints based on the past searches of the nearest neighbors to reduce the set of possible matching solutions. Our strategy was tested using public datasets and exhibited encouraging results.

*This work has been partially supported by Brazilian agencies CAPES and CNPq.

[1]Federal University of Rio Grande do Sul , Informatics Institute, Porto Alegre, Brazil {rcneuland; fcsrodrigues; dpittol; rqmaffei; mariana.kolberg; prestes}@inf.ufrgs.br

Fig. 1. Examples of pictures from UofA [3] (top) and GPW [4] (bottom) datasets presenting day (left) and evening (right) views of the same place.

The paper is organized as follows. We first present the related work about place recognition methods in Section II. Section III presents a view of the known world modeled as a set of intervals. Section IV presents our proposal, a visual place recognition method inspired by interval analysis theory. In Section V, we evaluate and discuss the method through the analysis of the experiments performed with two public datasets. Finally, in Section VI, we conclude and discuss future work.

II. RELATED WORK

An essential part of the visual place recognition is the scene description. The most common ways to represent a scene is using either a set of local features or the whole image information [5]. Each image can enclose hundreds of local features, which means that describing a scene and finding a match to each feature implies a high computational cost. Besides, using local features under changing conditions can deliver poor results, since they are easily missed when images suffer from blur effect.

On the other hand, using the whole image information to generate only one descriptor can be faster to describe and

Manuscript 136 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

search than using a set of features. However, changes in point-of-view can affect the quality of the global descriptor [5]. It is also possible to use raw images as descriptors [2], once computing the difference between two images is essentially subtracting the corresponding pixels of each other. Yet, this approach can be computational costly and strongly dependent on the point-of-view.

Different approaches to handle visual place recognition have been proposed using the most varied types of image representation. FAB-MAP [6] and SeqSLAM [2] techniques are examples of milestones in the state-of-art. They are based on bag-of-words [8] and sequence analysis respectively and have inspired new researches since then.

FAB-MAP is a probabilistic approach to place recognition based on place appearance. Inspired by Bag of Words (BoW) image retrieval systems [8], FAB-MAP converts the incoming sensory data into a BoW representation. The method relies on the fact that some words tend to appear and disappear together in specific combinations because they are present in the same objects. In spite of being a milestone for place recognition, it needs a training phase to create the BoW vocabulary and does not deal with environmental changes, once it is dependent on features similarity. FAB-MAP is suitable for applications at the scale of a few kilometers, but with a high computational cost. This fact motivated the elaboration of a new version called FAB-MAP 2.0 [9] which keeps the essence of its predecessor exploiting its characteristics to reduce computation and memory requirements.

On the other hand, SeqSLAM focuses on environments with perceptual change, comparing information collected in the same place but taken at different moments during a day, at different types of weather conditions or even at different seasons of the year. In unstructured environments or changing conditions, features may not be reliable enough to provide relevant information to describe a scene precisely. Thus, SeqSLAM considers sequences of images to match places. The authors claim that looking for matchings within sequences bring better results than analyzing single images only. Fast-SeqSLAM [3] is an enhanced version of the SeqSLAM. The method uses a tree structure to store the image descriptors and the nearest neighbor algorithm to speed up the search for matchings. Besides, it uses HOG [10] descriptor to represents and compare the images, differently of the SeqSLAM that uses image subtraction. The HOG descriptor uses the distribution of directions of gradients to represent an image. It has been used to describe features or, as shown in Fast-SeqSLAM, whole images. However, it is computationally expensive. In comparison to SeqSLAM, this new approach reduces the computational cost keeping the same results.

III. MODELING THE KNOWN WORLD USING INTERVALS

This section presents the known world modeled using interval theory and some operations essential to understanding of the proposed method.

A. *Interval-based modeling of robot observations*

In our approach, we consider a robot in a workspace $\mathbb{W}$ modeled as a Euclidian space $\mathbb{R}^n$, $n = 2$ or 3 [11]. The robot moves along a path defined as a continuous function

$$\tau : [0, 1] \rightarrow \mathbb{W}.$$

We can obtain a point p of the path through

$$p_e = \tau(e),$$

where $e \in [0, 1]$.

The robot creates the sets $\mathbb{U}$ and $\mathbb{P}$ while collects information about its surroundings. $\mathbb{U}$ is the set of all observations obtained and $\mathbb{P}$ is the set of points where the information was collected by the robot. The points in $\mathbb{P}$ are not necessarily linked to a known coordinate of the world. Figure 2 presents the relation between these sets defined by the function

$$\gamma : \mathbb{U} \rightarrow \mathbb{P}.$$

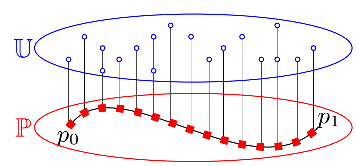

Fig. 2. Relation between $\mathbb{U}$ and $\mathbb{P}$.

Therefore, we can establish that given two observations $u, v \in \mathbb{U}$, there is a path between $\gamma(u)$ and $\gamma(v)$. Besides, we can sort the observations by the order in which they appeared. Thus, the following function labels each observation:

$$o : \mathbb{U} \rightarrow \mathbb{N}. \tag{1}$$

We define the o-distance between two observation as

$$\rho : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{N}.$$

For instance, given the observations u, $v \in \mathbb{U}$, the o-distance between them is

$$\rho(u, v) = |o(u) - o(v)|.$$

Be aware that the o-distance is related to the order that the robot collected the information; it does not consider the information contained in the observation itself.

The similarity measure between observations is another essential definition. This measure represents how much alike two observations are. The similarity function is defined by

$$\alpha : \mathbb{U} \times \mathbb{U} \rightarrow [0, 1].$$

Where α represents a generic similarity function. This measure is dependent on the kind of information we are comparing. When working with laser or sonar is common to use the sum of absolute differences, if the observations are images, the hamming distance between descriptors is often used to compute the similarity.

The definitions of o-distance and similarity when analyzed together may indicate that:

- the observations $u, v \in \mathbb{U}$ are from contiguous points of the path, if
$$\alpha(u,v) \geq t^0_\alpha \quad \wedge \quad \rho(u,v) \leq t^0_\rho,$$
where t^0_α and t^0_ρ are predefined thresholds used to indicate minimum similarity and maximum o-distance between observations collected in contiguous points.
- the observation $u \in \mathbb{U}$ was collected during a revisit when there is at least one $v \in \mathbb{U}$ where
$$o(u) > o(v) \ \wedge \ \alpha(u,v) \geq t^0_\alpha \ \wedge \ \rho(u,v) \geq t^1_\rho,$$
given that t^1_ρ is a predefined threshold used to indicate the minimum o-distance between observations during a revisit.
- the robot is not moving, or it is in a symmetric region when it collects a set of observations $\mathbb{U}' \subseteq \mathbb{U}$, and for all pairs $u, v \in \mathbb{U}'$ the following condition holds
$$\rho(u,v) < |\mathbb{U}'| \quad \wedge \quad \alpha(u,v) \geq t^0_\alpha,$$
where the order labels of the elements of $\mathbb{U}$ given by 1 are inherited by $\mathbb{U}'$.

Observations collected in contiguous points of the path, usually share a large amount of information. Thus, we can join these observations to represent a more significant region, creating a simplified version of the environment.

We can use o-distance and similarity definitions to create observation groups. Each group is a subset of $\mathbb{U}$ representing a region observed from the path and has an anchor element used as the base of comparison to create this subset. Given $\mathbb{G} \subseteq \mathbb{U}$, which has g as anchor element, an observation u is part of $\mathbb{G}$ if all following conditions are respected:

1) $u \in \mathbb{U}$;
2) $\alpha(u,g) \geq t^1_\alpha$, i.e. the similarity measure between g and u must be higher than a threshold t^1_α;
3) $\rho(u,g) \leq t^2_\rho$, i.e. the o-distance measure between g and u must be smaller than a threshold t^2_ρ;
4) the elements of $\mathbb{U}$, which have order labels between $o(g)$ and $o(u)$ belong to the set $\mathbb{G}$.

According to the conditions presented above, a set $\mathbb{G}$ can be represented through an interval $[g] = [g^-; g^+]$, where the infimum and supremum are defined by:
$$g^- \stackrel{\text{def}}{=} \min\{o(u) \ : \ u \in \mathbb{G}\}$$
$$g^+ \stackrel{\text{def}}{=} \max\{o(u) \ : \ u \in \mathbb{G}\}$$

The interval $[g]$ can be mapped to a subpath of τ, as presented in Figure 3.

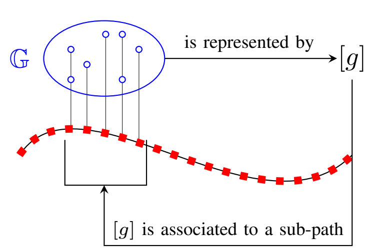

Fig. 3. A region of the environment represented by an interval.

B. Operations with intervals

Following are presented some interval operations necessary for the understanding of the method proposed in Section IV. Other interval operations can be seen in [7].

Considering the operator $\diamond \in \{+, -\}$ and the intervals $[x]$, $[y] \in \mathbb{U}$,
$$[x] \diamond [y] \ \stackrel{\text{def}}{=} \ [\{x \diamond y \mid x \in [x], \ y \in [y]\}].$$

For instance,
$$[x^-; x^+] + [y^-; y^+] = [x^- + y^-; x^+ + y^+].$$

It is important to highlight that a scalar number can be treated as a punctual interval. For instance, number 2 can be represented by the interval [2;2].

Interval approaches are known for generating guaranteed solution sets, which is true when the problem model and errors bounds are known. However, occasionally it is not possible to predict all noises of the system, and when this happens, the constraints lead us to an empty solution. To overcome that, we use relaxed approaches where not all constraints need to be satisfied to generate a valid solution.

The intersection operation is an example. Given two intervals $[x], [y] \in \mathbb{U}$, the intersection between them is defined by
$$[x] \cap [y] \stackrel{\text{def}}{=} \{x \mid x \in [x] \text{ and } x \in [y]\}.$$

Still, considering a set of intervals, even if there is none intersection among all intervals in the set, it may exist among a subset of them. The number of intervals intersecting can be adjusted according to the needs. This is called q-relaxed intersection [12]. Given a set of intervals $[x]_1, [x]_2, ..., [x]_n \in \mathbb{U}$, the q-relaxed intersection denoted by
$$\bigcap^{\{q\}} [x]_i \quad \text{for } 1 \leq i \leq n,$$
results in a set of all $x \in \mathbb{U}$ which belong to all $[x]_i$, except q at most.

IV. INTERVAL INSPIRED APPROACH TO PLACE RECOGNITION

We propose a novel approach to deal with the place recognition problem inspired by interval analysis theory. For comparison reasons, we adopt the same scenario presented by Milford and Wyeth [2]. The method works with two sets of images - a source and a query - from two traversals of the same environment extracted at different times. Given this scenario, the method aims to identify matching between images from the query and source sets.

A. Loading, resizing and describing images

We adopted a binary descriptor called LDB [13]; which was developed to be more robust, more distinctive and faster than other state-of-art binary descriptors. It divides the area of interest into a grid and computes the intensity and the first order gradient of each grid's cell. After that, it performs binary tests comparing the cells intensity and gradients among each other. LDB uses multiple granularities of grids capturing the scene structure with different levels of details. The LDB descriptor is composed of the concatenation of descriptions of all different level grids.

LDB can be used to describe a patch or a whole image. In our case, it is used as a global descriptor given its low memory requirements and fast comparison process based on Hamming distance. However, we propose a slight change in the way the image is described, i.e., each quadrant of the image generates a descriptor, which are concatenated to represent the image as one single descriptor.

Figure 4 presents the enhancing of the retrieval results obtained compared to describing the image using only one central keypoint. The x-axis shows the number of images selected per iteration; these images are the most similar to the query, i.e., the best matches. The y-axis presents the number of times that the correct retrieval was among the most similar images selected. We used the GPW [4] dataset with two different image sizes to perform the tests - 64×64 and 128×128 . Despite the extra computational cost, the improvements in the retrieval results are worthwhile. Thus, we use the configuration of 128×128 image size described with the modified LDB to perform the experiments in Section V.

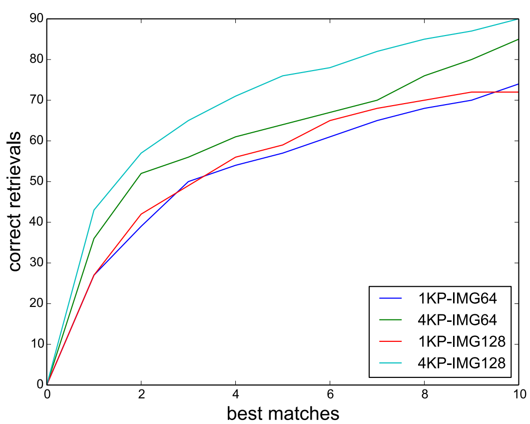

Fig. 4. LDB retrieval.

B. Modeling the world as a set of observation intervals

Our approach is based on the idea of modeling the known world as a set of intervals as shown in Section III. In our context, the set $\mathbb{U}$ corresponds to the source set, where the images are the observations. Each image has a numeric sequential label according to the order it was taken and it is represented by our modified LDB descriptor. The similarity between two images is measured based on the Hamming Distance of their descriptors.

Our method groups images according to their label order and similarity, holding the grouping requirements defined by the four conditions presented in Section III. The anchor element of a group is the one with the lower label among all images in the group. From these groups, the method creates a set of intervals representing the known world and a tree structure to store this set to speed up the search. It is worth noting that the searches are based on the anchor image of each interval.

C. Searching for matchings

The method loads, resizes and describes the images from the query set individually as the robot moves across the environment, and, one by one, they are processed. Our approach searches for matchings based on a sliding window of past iterations. The size of this window grows, starting at the moment the first query image is loaded until a predefined upper bound w. Our method search for matchings by following the next steps:

- *Robot motion detection*: The information about the robot motion is binary; we assume either the robot is moving or not. For that, the method tests the similarity between the last loaded image and its four predecessors, if it is higher than a given threshold t^2_α, we consider the robot is not moving. This information is stored in a one-dimensional matrix $M_{w \times 1}$ to be used in the next steps computation.
- *Finding interval matchings*: our method uses the descriptor of the current query image to search for k interval nearest neighbors. These intervals represent regions of the environment that are most probably to contain the place where the query image was taken.
  Each interval nearest neighbor has two values associated, the weight z and the cumulative weight z^*. Being $[x]$ an interval nearest neighbor, its weight $z_{[x]}$ is given by the similarity between the anchor element of $[x]$ and the query image. The cumulative weight $z^*_{[x]}$ is initialized with the same value of $z_{[x]}$ and updated in the weighting step. The method keeps the nearest neighbors in a matrix $N_{w \times k}$ of intervals, each line of N stores the nearest neighbors of a query. This information will be essential to compute the next steps.
- *Biased intervals*: The same interval $[y]$ being repeatedly selected in near iterations implies that the robot is not moving or it is crossing some symmetric region. If that is not the case, i.e., the robot is moving, this interval is called biased, and our method penalizes it by
$$z_{[y]} = z_{[y]}/2.$$
- *Intervals propagation*: Each interval of N is propagated according to the robot motion stored in M.
  Each interval nearest neighbor in line l of the matrix N is added to a scalar value m_l and grouped in a set $\mathbb{A}_l$:
$$\mathbb{A}_l = \bigcup_{c=0}^{k} \Big(N_{l,c} + m_l \Big).$$
  Where m_l represents the robot motion between the moment the interval was selected and the current iteration. m_l is based on the information in M, and it is defined by
$$m_l = \sum_{i=l+1}^{s} M_i.$$

Manuscript 136 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

Where s is the number of lines in M. Thus, the set $\mathbb{A}_l$ contains the intervals from line l of the matrix N after being affected by the robot motion. Now, the method joins the sets computed to each line in a set $\mathbb{C}$, which represents possible matchings to the current image query according to past iterations:

$$\mathbb{C} = \bigcup_{l=0}^{s} \mathbb{A}_l.$$

- *Weighting*: The intervals in the set $\mathbb{C}$ represent regions that were considered as a possible match at least once in past iterations. After the intervals propagation step, each interval indicates where would the current match be. Equal intervals are linked to the same region and can be deleted without losing information. Therefore, duplicated intervals from set $\mathbb{C}$ are unified and their weights are evaluated. Suppose the intervals $[u]$, $[v] \in \mathbb{C}$ are equal. If

$$z_{[u]} > z_{[v]},$$

$[v]$ is deleted from $\mathbb{C}$. However, the region they represent needs to be emphasized, which is done by updating the cumulative weight of $[u]$:

$$z^*_{[u]} = z^*_{[u]} + z_{[v]}.$$

Lastly, the cumulative weights are normalized.
- *Matching region*: Intervals represent regions of the world, and we need to find the most similar one to the current image query. Assuming that set $\mathbb{C}$ contains the solution, our method selects the intervals with the highest cumulative weight to search for the current matching. These intervals are input to the q-relaxed intersection presented in Section III, where the q parameter to the intersection is the smallest possible that returns a non-empty solution. The result of the relaxed intersection defines the current matching region.
- *Final matching*: After the definition of the interval matching, the method sweeps the images represented by the interval computing the similarity between each of them and the query image. The weighted average defines the final matching.

V. EXPERIMENTS

In this section, we present the results of the performed experiments, comparing the proposed approach with OpenSeqSLAM [14] that is an implementation of the SeqSLAM method. For evaluating our approach, we use two public datasets, which have ground truth information available. More details about the datasets can be seen in Table I.

Table II presents the parameters used in the experiments. These parameters were chosen to maximize the effectiveness of the method.

First, we present the matchings obtained by the methods using each dataset. Ideally, the perfect matching is represented by a full diagonal from bottom-left to up-right. In both datasets, a considered successful matching is at a maximum of three images far from the ground truth.

TABLE I
DATASETS

Name	Description
UofA [3]	University of Alberta dataset, from the University of Alberta, Edmonton, Canada. The dataset was collected on Campus at two different times of a day with a Husky robot.
GPW [4]	Garden Points Walking dataset, from the Queensland University of Technology, Brisbane, Australia. The dataset has two traversals, one collected during the day and the other at night.

TABLE II
PARAMETERS

Symbols	Description and Values
t^1_α	minimum similarity with the anchor to be in the same interval. $t^1_\alpha = 0.85$
t^2_ρ	maximum interval size, here we use the dataset size. $t^2_\rho = 200$ to GPW and $t^2_\rho = 645$ to UofA
w	maximum number of past iterations analyzed at each iteration. $w = 200$
k	number of nearest neighbors to each query. $k = 10$

Figure 5 shows the matchings result for both methods, ours (left) and OpenSeqSLAM (right) on UofA dataset. In this figure, it is possible to see that our method has a better result than the OpenSeqSLAM, being very close to the full diagonal.

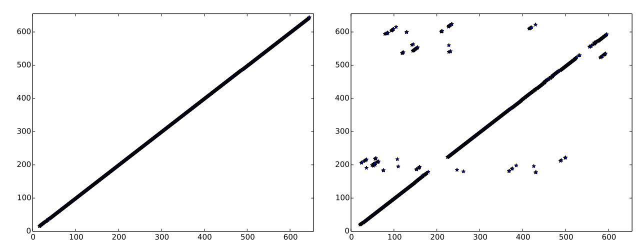

Fig. 5. Matchings of our method (left) and OpenSeqSLAM (right) to UofA dataset.

Figure 6 shows the same kind of result to GPW dataset. Thus, our method obtains more true matchings than OpenSeqSLAM for both datasets.

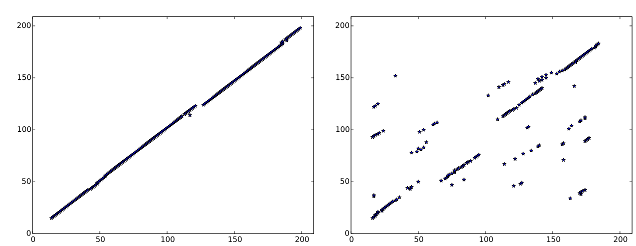

Fig. 6. Matchings of our method (left) and OpenSeqSLAM (right) to GPW dataset.

Since our method uses past information to estimate the current matching, it needs to wait for some iterations at the beginning of the computation to calibrate itself, to the presented datasets we used 15 iterations. This calibration time explains the lack of matching results we can see on the bottom-left corner of Figures 5 and 6.

Manuscript 136 submitted to 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 1, 2019.

OpenSeqSLAM also waits for some iterations before given matching results, and according to the available code, it doesn't compute the matchings to a set of images at the end of the dataset also. This occurs because it uses a sliding window to compute the matching to a central window element, i.e., it uses information of future queries.

Figures 7 and 8 present the precision-recall curves to the results obtained with UofA and GPW datasets respectively. Each figure compares the proposed approach to OpenSeqSLAM.

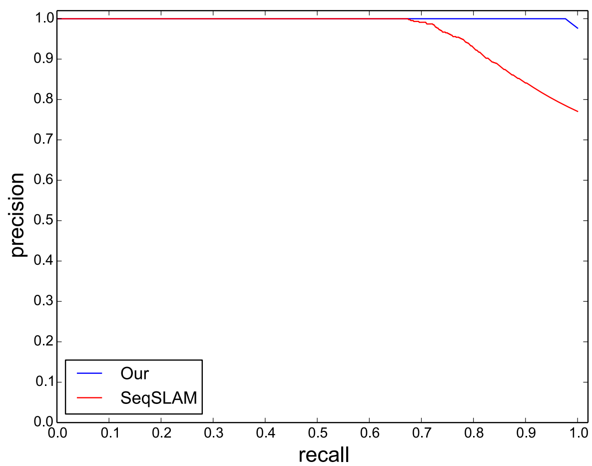

Fig. 7. PR - UofA dataset.

The precision-recall curves show that our approach brings better results than OpenSeqSLAM to both datasets. It is important for place recognition to evaluate the recall obtained at 100% of precision. This metric is relevant in case of dealing with SLAM algorithms, where an incorrect matching for a loop closure can cause massive mapping failures. At 100% precision, our method achieves a recall of 97% on the UofA dataset, and 92% on the GPW dataset compared to 67% on UofA and 23% on GPW using OpenSeqSLAM.

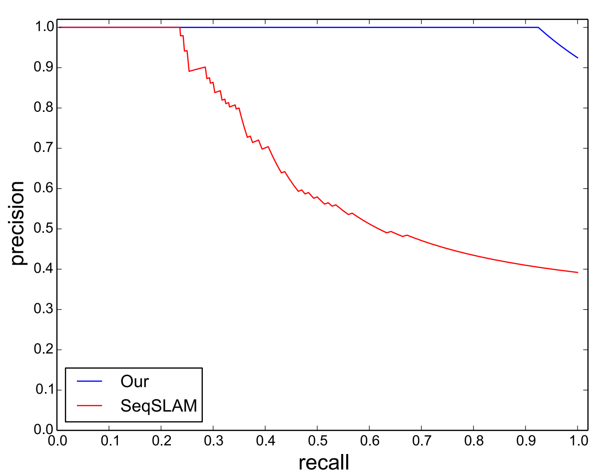

Fig. 8. PR - GPW dataset.

VI. CONCLUSIONS

We propose a visual place recognition method inspired by interval theory using a monocular camera. Apart from modeling the known world as a set of intervals, it relies on a relaxed set of constraints based on the search of nearest neighbors from past iterations.

The method shows high success rate finding matchings even in datasets with significant perceptual changes. We optimize the search for matchings by computing many steps based purely in the interval information, i.e., using the order in which the observations were taken, without image comparison. Furthermore, we presented a different way to use LDB as a global descriptor, which enhanced the images retrieval rate.

The tested datasets provide one image per second, and our method is capable of producing a matching result in less than one second in an average computer (Intel Core i7 - 8GiB RAM). An accurate analysis of its computational cost is part of the future work. Besides, our approach doesn't use information corresponding to the future of the current query. These characteristics indicate the potential of the method for online tests.

Also, as future work, we intend to improve the motion estimation to work with datasets with high variability in speed.

REFERENCES

[1] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Fast loop-closure detection using visual-word-vectors from image sequences," *The International Journal of Robotics Research*, p. 0278364917740639, 2017.

[2] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), Inter. Conf. on*. IEEE, 2012, pp. 1643–1649.

[3] S. M. Siam and H. Zhang, "Fast-seqslam: A fast appearance based place recognition algorithm," in *Robotics and Automation (ICRA), Inter. Conf. on*. IEEE, 2017, pp. 5702–5708.

[4] A. Glover, "Datasets," 2014, data retrieved from Robotics@QUT, https://wiki.qut.edu.au/display/cyphy/Datasets.

[5] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[6] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[7] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer, 2001.

[8] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision (ICCV), Inter. Conf. on*. IEEE, 2003, p. 1470.

[9] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR), Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[11] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic, 1991.

[12] Q. Brefort, L. Jaulin, M. Ceberio, and V. Kreinovich, "Towards fast and reliable localization of an underwater object: an interval approach," *Journal of Uncertain Systems*, vol. 9, 2015.

[13] X. Yang and K.-T. Cheng, "Ldb: An ultra-fast feature for scalable augmented reality on mobile devices," in *Mixed and Augmented Reality (ISMAR), Inter. Symposium on*. IEEE, 2012, pp. 49–57.

[14] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, Inter. Conf. on Robotics and Automation (ICRA)*. IEEE, 2013, p. 2013.