International Journal of Engineering & Applied Sciences (IJEAS)
Vol. 12, Issue 1 (2020) 1-14
http://dx.doi.org/10.24107/ijeas.694523
Int J Eng Appl Sci 12(1) (2020) 1-14

Matching Image Sequences using Mathematical Programming: Visual Localization Applications

Abdul Hafiz ABDULHAFIZ

Department of Computer Engineering, Hasan Kalyoncu University, Gaziantep, Turkiye
E-mail address: abdul.hafez@hku.edu.com

ORCID numbers of authors:
0000-0002-1908-5521

Received date: 25.02.2020
Accepted date: 23.04.2020

Abstract

Visual localization of a robot is to determine the location using visual input from a camera. We present in this article a new visual localization algorithm to find the robot location with respect a visual route map stored as a video sequence. The sequence of the current and past images is matched to the map, i.e. the reference image sequence, to produce the best match of the current image. The image sequence matching is achieved by measuring the similarity between the two image sequences using the dynamic time warping (DTW) algorithm. The DTW algorithm employs Dynamic Programming (DP) to calculate the distance (the cost function) between the two image sequences. Consequently, the output of the alignment process is an optimal match of each image in the current image sequence to an image in the reference one. Our proposed DTW matching algorithm is suitable to be used with a wide variety of engineered features, they are SIFT, HOG, LDP in particular. The proposed DTW algorithm is compared to other recognition algorithms like Support Vector Machine (SVM) and Binary-appearance Loop-closure (ABLE) algorithm. The datasets used in the experiments are challenging and benchmarks, they are commonly used in the literature of the visual localization. These datasets are the" Garden point", "St. Lucia", and "Nordland". The experimental observations have proven that the proposed technique can significantly improve the performance of all the used descriptors like HOG, LDB, and SIFT. The performance of these features is compared to the case of using the proposed DTW instead of the classical nearest neighbor. In addition, it was able to the SVM and ABLE localization algorithm.

Keywords: Visual localization, Image sequence matching, Dynamic programming.

1. Introduction

The visual localization techniques belong to the content-based image retrieval algorithms. It can be used for visual localization, also called Place recognition, by using the available reference sequence (experience(s) of the robot) to determine the appropriate response for the current observation. In addition, the significant improvements in the visual localization topic, leads to increase the attention of the robotics community [1-4]. Fig. 1 depicts the main components of the visual localization process. The main two components are the feature extraction and the localization algorithms. We propose in this work a localization algorithm that is based on DP, and test it with different feature extraction methods.

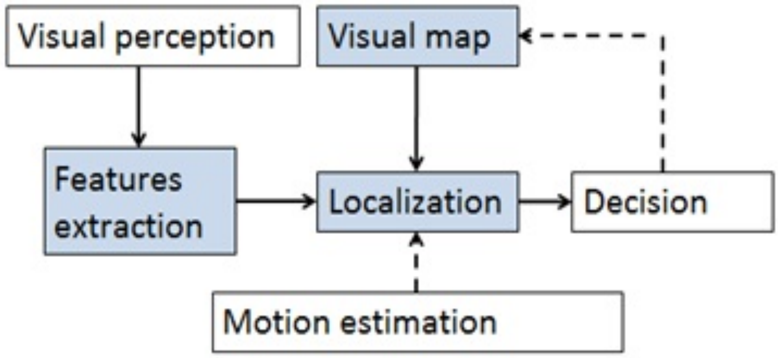


Fig. 1. Schematic diagram of the visual localization (place recognition) process showing the different components and links between them.

There are several works in the literature on visual localization and place recognition. They mainly focus on matching two or more image sequences for the purpose of localizing the object in the environment. The work called FabMap that is presented in [5], SeqSLAM presented [1], the Flow networks based algorithm presented in [6], and ABLE [7], [8] are example of the pioneering works in the topic of visual localization.

Several attempts on localization with respect to visual map that is represented using sequences of images are reported in the literature. These attempts include visual localization in crowded environment [9], [10], experience based navigation for long-term localization [11], [12], and summarizing the map of experiences for long-term localization [13]. Other methods like RTAB-Map [14], [15] focus on memory management to satisfy the real time conditions.

The contribution of this paper is mainly a novel visual localization algorithm that uses image sequence alignment for localizing mobile robots. In this work, the sequence matching and alignment is done using the Dynamic Time Warping (DTW). In addition, a set of common handcrafted features used to compare the proposed algorithm with two of the-state-of-the-art algorithms. These are the ABLE and SVM ones. The validation process for the developed approach and the experimental study were performed using some common benchmark datasets.

2. Review of Previous Works

The RTAB-Map presented in [14], [15] works fine with the loop closure problem solving issues like when the map is large enough to slow the system down. RTAB-Map track SURF features using discrete Kalman filter without dealing with processes of matching and retrieving multiple images which are the core of our proposals. Our main focus, in contrast, is on localization in environments that show considerable changes in its representation.

FabMap localization algorithm system [16] is topological pure image retrieval that use probabilistic model to update the vocabulary of the image retrieval. Image retrieval techniques are borrowed from computer vision community [17] in which a set of selected features are stored in a dictionary of vocabularies, while the reference images are sorted using inverted index techniques. In particular, the inverted index is updated to accommodate new positively observed features. FabMap is a data driven approach to calculate the observation likelihood. It uses a bag-of-word of SIFT features model to describe the distinctiveness of each feature during the learning stage. There still is a scalability problem with bag-of-words model since it needs large number of vocabularies for larger environments.

Visual localization in crowded environment from multiple experiences was proposed in [9], [10]. The work addresses the case where the vehicle frequently visits the same environment updating it's a priori knowledge after each visit. A bag-of-words model along with a weighted inverted index are employed as a map of the considered environment. The aim was to learn useful visual features and visually stable features that are static in the environment by assigning a weight to each feature. In this work, the visual map is suggested to store several reference image sequences since cyclic appearance changes require more than one representation model.

Long-term localization is achieved based on visual experience as presented in [12]. The visual experiences here are stored as a series of visual odometry data attached to a sequence of visual images. A new experience is created and stored after each visit to non-recognized place. This is noted as failure of all localizer which are associated with the multiple experiences stored in the map. The experiences are connected to each other via GPS priors or geometric consistency among image frames. However, they do not systematically explore the connectivity between experiences. Hence, a few nodes are connected in each experience depends on whether an accurate GPS data is available or not for example.

Multiple experiences case is also considered in [11]. Here, a probabilistic cost map is created in a self-supervised manner using a Gaussian process. Multiple representations of the environment in the visual map are used here to describe the travers-ability of the environment rather than to localize the robot. The Gaussian process model is successfully used to plan a path for the robot through traversable areas. A summary map is proposed in [18] as a solution for the mapping from multiple sequences problem, in which scoring functions are used, similarly to [9], [10], to update the summary map after every arrival of new experience. The scoring function is used to evaluate the usefulness of the landmarks and trajectory segments. Those have high score are retained in the summary map while others are forgotten. However, these methods represent fine-grained information about the environment, but do not consider any appearance aspects since it does not store whole image information, that what is done in our proposal.

FabMap considers the localization problem as a visual loop closure problem. On the other hand, the SeqSLAM algorithm [1] using and searching all possible visual maps matching to solve the localization problem. Overall, the variations of view and camera poses can badly affect the performance of such a technique. A recent impressive visual localization using sequence matching and consider the problem as an alignment of two sequences of images were presented in [6]. This approach works on minimizing the cost of the high computational complexity flow network. In more details, it uses the ABLE algorithm [7], to represent the sequences of images as binary code, then, the effect of changes in appearance while taking the images was reduced using the Illumination invariance color.

Recently, SVM was able to archive a good performance as a classifier in many filed such as object classification, image matching, etc. In [19], some vision-based techniques for visual place recognition were introduced. First, an image salience generation was adapted to improve the single image-based matching. Then, the Support Vector Machines (SVMs) was used to filter the outliers from both the reference and test datasets. In [20] the Kernel Principal Component Analysis (KPCA) was used to extract the image features. In more details, the SIFT features from a given image is extracted, then, the minimum Euclidean is used to find the distance between the extracted features and the visual codebook that was constructed offline by K-means. In the approach of [20], SVM was used for data analysis as a classifier. In [21], both HOG and LBP were used for visual localization.

3. Dynamic Programming and Dynamic Time Warping

DP aims at solving the problem in hand by combining the individual solutions to smaller problems that are sub problems of the original problem [22]. DP efficiently solves such subproblems when these subproblems overlap. In other words, when these subproblems are sharing another lower level subproblems like subsubproblems, and so on. A DP algorithm provide a solution to each of these subsubproblem, which is stored in a table matrix to avoid solving the same individual problem again.

The DTW algorithm has applications in several areas. Its applications are increasing since it was first introduced and developed in the 60s and 70s of the last century till today. Its application was initially explored to solve the speech recognition problem [23], [24], but it currently has applications in topics like: online and offline handwriting and signature verification [25], [26], sign language recognition [27], mining and searching databases of time series [28], [29], image understanding and computer graphics [30], surveillance [31], matching protein sequences in bioinformatics [32], and music applications [33].

DTW algorithm has become much popular his efficient solutions to the time series problems. It measures the similarity between two sequences by minimizing a cost function in order to detect their shape similarity over different times, i.e. they have different phases.

Let us have two time sequences $X = (x_1, x_2, \dots, x_n)$, where $n \in N$ and $Y = (y_1, y_2, \dots, y_m)$, where $m \in N$. Here, N is the group of natural numbers. DTW produces an optimal solution with a time complexity in the order of $O(MN)$. The data sequences must be sampled evenly at a uniform basis, so sometimes a resampling stage in the feature space is needed.

To compare these two different sequences X and Y using DTW, we need to use a local distance measure between two values x_i, y_j each from one sequence. This distance can be defined as a function $D(x_i, y_j) \in R$, where R is the real numbers. The value of the distance function between two elements from the sequences X and Y is smaller when they are more similar, and is larger when they are more different. Since DTW is a Dynamic Programming based algorithm, it is more convenient to call this distance function as the "cost function". Consequently, the process of finding the optimal alignment between the two sequences is becoming the arranging of the two sequences by optimizing the cost function.

3.1. Sequence Alignment using DTW

In this subsection, the mechanism of aligning the main two image sequences is explained. One of the sequences is $Y = \{y_j\}$ that is the reference one, here $j = (1, \dots, m)$, and the second sequence is the test that denoted by $X = \{x_i\}$ where $i = (1, \dots, n)$, this process uses the DTW, and as a first step, we build the cost matrix C. In general, as shown in Equation (2), the elements of the distance matrix are accumulated to obtain the cost matrix. Note that n represent the lengths of test sequence and m the lengths of the reference sequence.

In DTW, matrix C refers to the cumulative distance, i.e., $D(\mathrm{i}, \mathrm{j}) + Mcd$, where $D(\mathrm{i}, \mathrm{j})$ is the distance between currently matching two images. The $Mcd(\mathrm{i}, \mathrm{j})$ is the smaller distance $D(\mathrm{i}, \mathrm{j})$ among the images in surrounding neighborhood. It can formulated as in Equation (1):

$$Mcd(\mathrm{i},\mathrm{j}) = \min\begin{Bmatrix} D(\mathrm{i}-1,\mathrm{j}) \\ D(\mathrm{i},\mathrm{j}-1) \\ D(\mathrm{i}-1,\mathrm{j}-1) \end{Bmatrix}, \tag{1}$$

Then, the cost matrix C is filled by dynamic programming by implying the following relation:

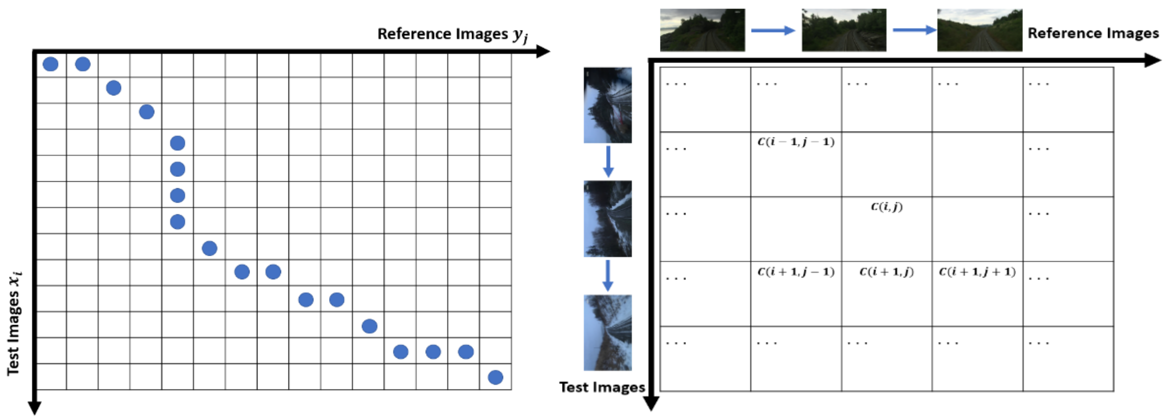
$$C\ (\mathrm{i},\mathrm{j}) = D(\mathrm{i},\mathrm{j}) + Mcd(\mathrm{i},\mathrm{j}) \tag{2}$$

After filling the matrix C, the optimal path $P = \{p_k\}$, where $k = (1, \ldots, L\ )$ is found by DTW. Hence, whenever images $x(i_k)$ and $y(j_k)$ are found to be a part of the optimal path P, they are represented by $p_k = (i_k, j_k)$.

It is worth mentioning that by back-tracing the matrix C and choosing the lowest cumulative distance of the previous elements we define what is called the optimal path. Hence, by minimizing the following function, we can obtain the optimal path, i.e., the path through the elements of the matrix C . These matrix elements have the minimum accumulated cost values $C\ (\mathrm{i},\mathrm{j})$.

$$\mathrm{Q(P)} = \sum_{l=1}^{L} C(i_l, j_l) \tag{3}$$

For more details about DTW, the reader is referred to [24], [34] and [35]. Fortunately, even after integrating the DTW, the complexity of our approach is reasonable as compared with other existing approaches and as compared with the improvement that has been archived by integrating the DTW. In other words, complexity equals $O(m \times u)$, whereas mentioned before m is the lengths of the reference sequence and u is the number of images in the optimal path. As a result, since u is much less than n, DTW decreases the number of required comparisons. However, the complexity of Flow network algorithm, the well common another matching algorithm, is estimated as $O(n \times m)$. In more details, the redaction in the complexity came from the sequential nature that applied by DTW, i.e., when a test image arrives let say $x(i)$, it will be compared with the sequence of reference images which has the same or lager index (j, where j >= i). It is worth mentioning that in this work the vehicle can move forward only, and allowing the vehicle to other directions can be done as future work. The diagram of the proposed sequence method is shown in Fig. 2.



(a) (b)

Fig. 2. This figure shows calculating the cost matrix element C (i, j) for matching mage xi with reference image y_j in (a), while the corresponding optimal matching path is shown in (b).

4. Sequence Matching Method for Visual Localization

In general, image matching/alignment has been used frequently for visual place recognition, however, its efficiency and execution time can be effected by the size of the used visual dataset and it is impractical for processing large visual datasets. In this paper, we have solved this problem by integrating the DTW into the image matching/alignment [24].

4.1. Image Features Presentation and their Distances

Three of the most popular handcrafted descriptors are used. These are the local difference binary features (LDB), the scale-invariant feature transform (SIFT), and the histogram of gradient (HOG), however, any other descriptor(s) can be integrated into the proposed algorithm. The used descriptors is summarized in the following:

SIFT: This descriptor work on detecting the image's key points which can be considered as the most important regions. Then, the appearance of these key points is characterized by a 3-D spatial histogram.

HOG: This descriptor starts by dividing the image into small squared cells, where each cell is then represented by a histogram of oriented gradients. Then the block-wise pattern is used to normalize the results obtained from the previous step.

LDB: This descriptor represents each image by a binary string. It firstly extracts the patches from the image. After that, the differences in the gradient and intensityare tested used to find the binary string. This process is done for a pairs of grid cells for each patch.

Based on the above, the distance between the feature vectors are given as

$$D(\mathrm{i,j}) \;=\; Distance(\mathrm{xi,yj}) = 1 - \frac{|Ax_i| \;\; . \;\; |Ay_i|}{\|Ax_i\| \;\; . \;\; \|Ay_j\|} \tag{4}$$

Here, the vectors Ax_i and Ay_i are extracted from images x_i and y_j. Also, the element $D(\mathrm{i,j})$ represents the distance between matching the test image x_i and the reference image y_j.

It is worth mentioning that related to the LDB descriptor, as it produces a binary vector, it has been proven that it is preferred to use Hamming distance with LDB. Hence, we have used the Hamming distance to represent the distance matrix $D(\mathrm{i,j})$ whenever LDB is used in this study, Hamming distance is given as

$$D(\mathrm{i,j}) \;=\; Distance(x_i, y_j) = Ax_i \oplus Ay_i \tag{5}$$

Finally, independent of the selected cost function, as soon as the elements of $D(\mathrm{i,j})$ are calculated, the same steps will be allows used by DTW for the alignment.

5. Experimental Study

Several experiments have been conducted in order to evaluate the performance of the proposed algorithm using different data sets. They are namely “St. Lucia” [36], “Nordland" [37], and the “Garden point" dataset. We compare the performance of the proposed algorithm with two of the recent similar works, they are SVM and ABLE in particular. Besides, the LDB, SIFT, and HOG handcrafted features are used in these experiments.

5.1. Datasets and Evaluation

The “St. Lucia" dataset has been recorded using a webcam that is fixed to a car along a selected route across the St. Lucia suburb. The dataset was collected during ten runs. The route was visited five times in the early morning and the afternoon to show the difference in appearance. The same visits are repeated later after two weeks.

The Nordland dataset contains four video sequences each of which is recorded using a camera fixed to a train during its 10 hours trip during a train journey in a different season. Garden point dataset consists of three series of images collected at the Queensland campus. Two of the series (Day Left and Day Right) were collected at day, but with slightly different viewpoint, the third one (Night Right) collected at night with the same pose of the (Day Right). Each of these series consists of 200 images where the labels of images represent the correspondence between the series.

The performance of our proposed DTW algorithm is analyzed and quantified using the precision-recall curve (PRC). The curve is obtained after finding the best matches between the current and reference sequence, and then calculate the number of true positives, false positives, and false negatives. Then, the precision is according to

$$P = \frac{TP}{TP+FP}, \tag{6}$$

and the recall value is given as

$$R = \frac{TP}{TP+FN} \, . \tag{7}$$

In our analysis, a positive match is considered when the distance $D(\mathrm{i}, \mathrm{j})$ is smaller than a given threshold t. If the distance is larger, we consider the match as a negative match.

5.2. Experiments and Results Analysis

In the following subsections we present our experiments outcomes. The three algorithms, i.e. DTW, ABLE and the SVM are evaluated using three types of features which are the HOG, SIFT, and LDB binary features. In these experiments, we have resized the image frames to a single unified dimensions. The size of the grid of HOG feature descriptor is set to 32 x 32, while a 128 bins are used for representing SIFT descriptor and a total of 40 SIFT features points are extracted. In addition, the OpenCV library was used to implement DTW and HOG, SIFT descriptors with python programming language. The source code provided by OpenABLE [1] was used to obtain the LDB features.

5.3. Experiments using “Nordland” Dataset

We present in this section the experimentation using the “Nordland" dataset, where the “Summer” and “Winter” sequences were used in these experiments. Two experiments were mainly conducted using this dataset.

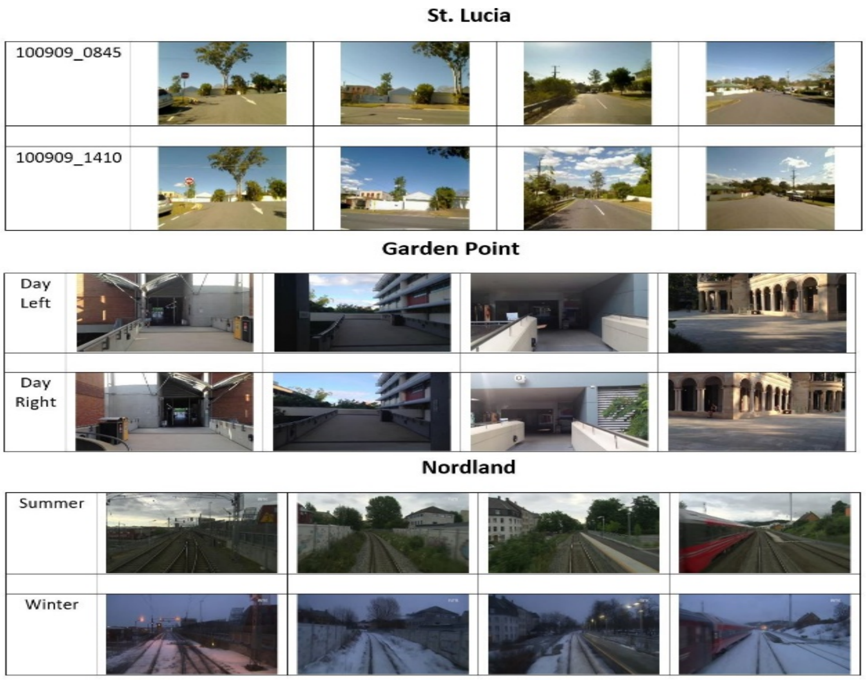


Fig. 3. Examples of pair images from the three datasets used in our experimentation. They are the "St. Lucia", "Garden point", and "Nordland" datasets from top down.

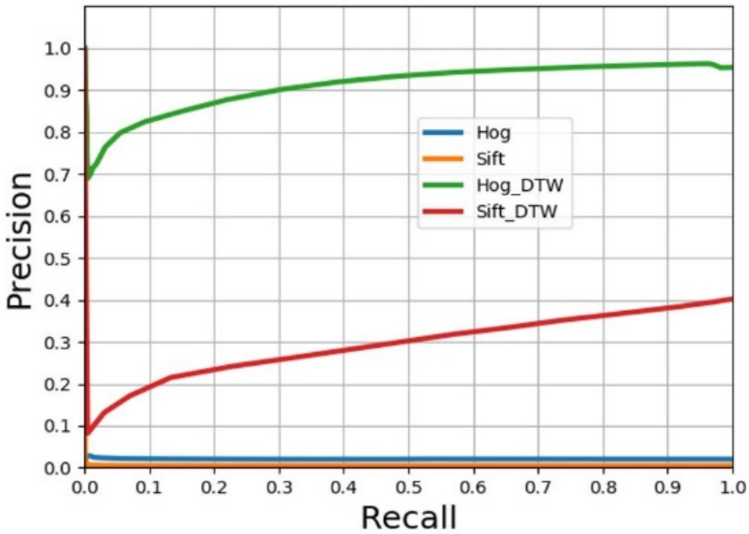


Fig. 4. Using the "Nordland" Dataset, the performance of HOG and SIFT using a nearest neighbor model is compared to the one using the, HOG-DTW and SIFT-DTW, i.e. using the proposed DTW. The precision-recall curve is used to quantify the performance.

5.3.1. Experiment 1

The performance of using the DTW algorithm along with HOG and SIFT was explored in this experiment using the "Nordland" dataset. The PRC is found using SIFT and HOG in a nearest neighbor model. In other words the best similar HOG or SIFT feature vector is used to classify the image from the test sequence to its matching frame from the reference sequence. After that the curve is found by applying the DTW to the image sequences represented by the HOG and the SIFT descriptors. Fig. 4 shows these PRCs.

As shown in Fig. 4, using the DTW algorithm has shown higher precision values for all recall values. The threshold between true or false positive matches is set to 1 frame.

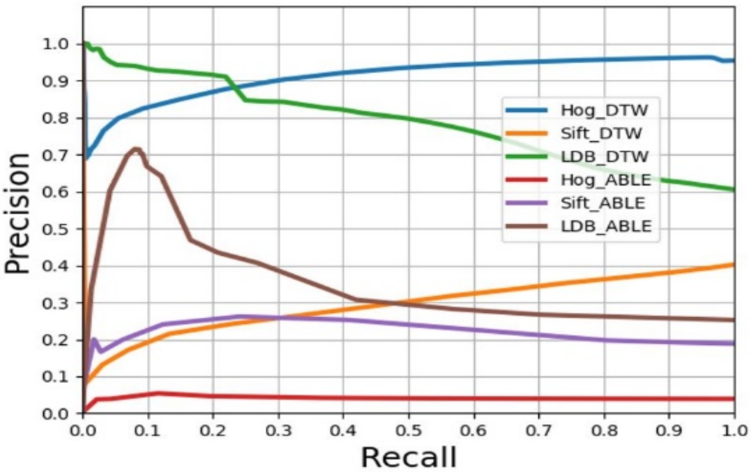


Fig. 5. Using the “Nordland" Dataset, the performance of DTW and ABLE algorithms is compared using the three types of features, HOG, SIFT, and LDB. The precision-recall curves is used to quantify the performance.

5.3.2. Experiment 2

The second experiment evaluates the performance of our proposal with respect to the ABLE algorithm using this “Nordland" dataset. As depicted in Fig. 5, both the HOG and LDB features using the proposed DTW (in green and blue) have outperformed the SIFT features, and also outperformed the same features using the ABLE algorithm. However, the HOG features have shown higher precision value for medium and higher recall values.

5.4. Experiments using “St. Lucia" Dataset

The experiments in this section are carried out using the “St. Lucia" dataset. The reference sequence was recorded at 8:45 o’clock morning, while the test sequence was recorded at the afternoon from the same day. Since the frames are tagged with its GPS coordinates values, a 15 meters threshold is selected to discriminate true positive matches from false positive ones.

Two experiments have been conducted using this dataset. The performance of the DTW algorithm is compared to the one of the ABLE algorithm and the SVM one.

5.4.1. Experiment 1: DTW vs. ABLE

The precision-recall curves resulted from these comparison with ABLE algorithm are shown in Fig. 6. The figure shows that the ABLE algorithm has resulted with a lower precision, regardless the recall values is large or small. This low precision has been observed for the different three feature descriptors. Besides, the precision of DTW while using the SIFT descriptor was small as well. The proposed DTW algorithm has shown a meaningfully higher precision for both descriptors HOG and LDB.

5.4.2. Experiment 2: DTW vs. SVM

Fig. 7 shows the comparison with SVM. The Figure shows lower precision using SVM algorithm. While the precision of DTW with SIFT descriptor was low, it achieved a

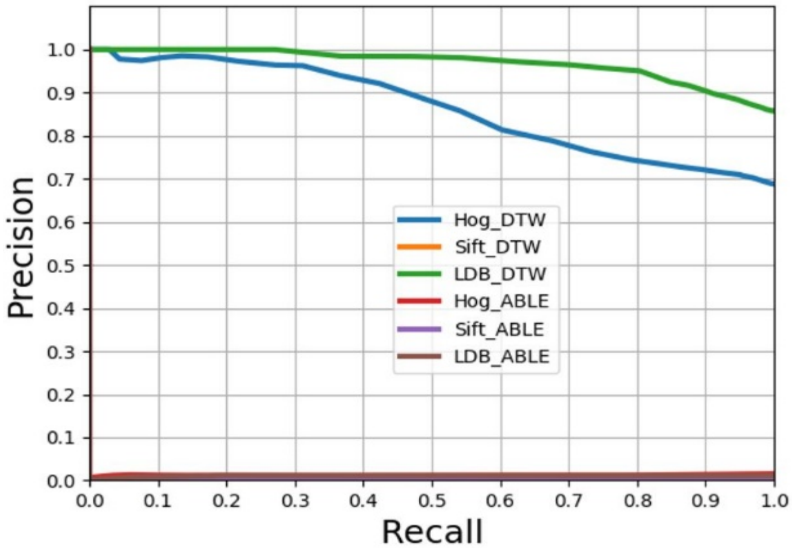


Fig. 6. Using the “St. Lucia" dataset, the performance of DTW and ABLE algorithms is compared using the three types of features, HOG, SIFT, and LDB. The precision-recall curves is used to quantify the performance.

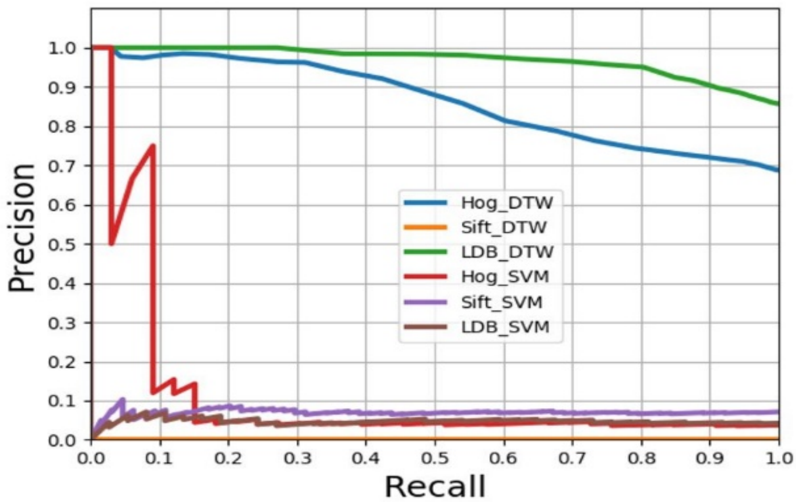


Fig. 7. Using the “St. Lucia" dataset, the performance of DTW and SVM algorithms is compared using the three types of features, HOG, SIFT, and LDB. The precision-recall curves is used to quantify the performance.

significantly higher precision for both HOG and LDB descriptors. It can be concluded that DTW with HOG and LDB has the best results.

5.5. Experiments using “Garden point” Dataset

The performance of our algorithm is compared to both ABLE and SVM algorithms using the “Garden point" dataset. The performance is quantified in the form of the precision-recall curve as depicted in Fig. 8 for comparison with ABLE and in Fig. 9 for comparison with SVM.

Similar to the case with “St. Lucia" dataset, the proposed DTW algorithm has outperformed both the ABLE and SVM algorithms, as depicted in Fig. 8 and Fig. 9 respectively. In these two figures, the precision-recall curve shows higher precision using the proposed DTW algorithm for all recall values. It is worth to notice that DTW has resulted with relatively lower precision values using the “Garden point" dataset than the case of using the “St. Lucia" dataset. This is

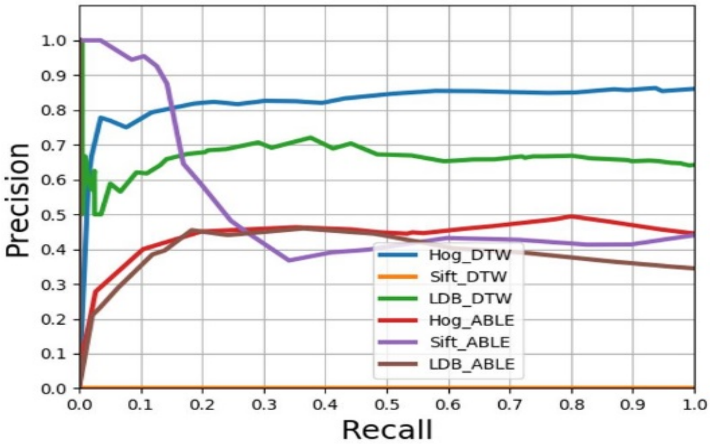


Fig. 8. Using the “Garden point" dataset, the performance of DTW and ABLE algorithms is compared using the three types of features, HOG, SIFT, and LDB. The precision-recall curves is used to quantify the performance.

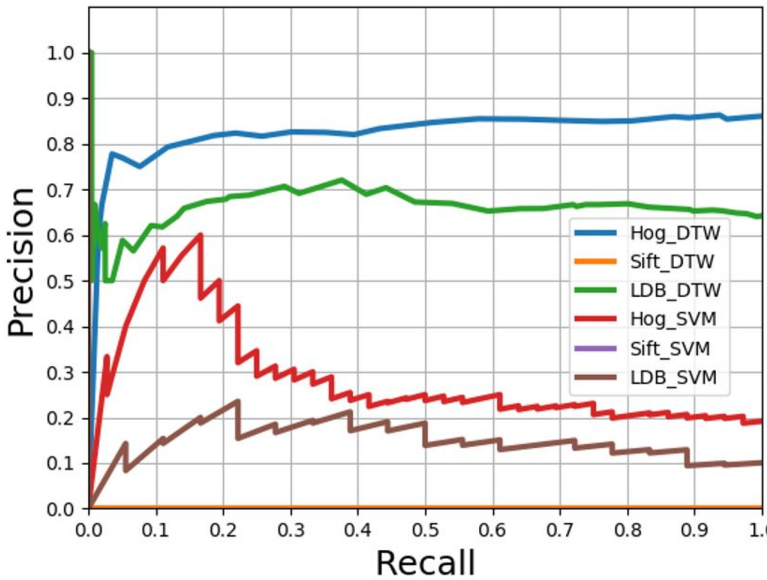


Fig. 9. Using the “Garden point" dataset, the performance of DTW and SVM algorithms is compared using the three types of features, HOG, SIFT, and LDB. The precision-recall curves are used to quantify the performance.

due to the nature of the “Garden point" dataset since it is highly textured comparing to the `St. Lucia" dataset. In addition to the different in pose and illumination between the test and reference sequences.

6. Conclusion Remarks

The dynamic programming based DTW algorithm is used here to perform visual localization of an autonomous agent. The basic idea of the contribution in this paper is to achieve a visual localization by aligning the pre-knowledge of the agent about the environment, i.e. stored in the form of a previously observed sequence of images, to the currently observed sequence of images.

The matching is achieved by dynamic time warping. The evaluation results have shown higher precision values for most of the recall values while using the proposed DTW algorithm. This proves the superiority of it with respect to the ABLE and SVM algorithms.

We plan to explore deep learning features with the proposed algorithm. In addition, a future work may include the introduction of encoding stage to reduce the complexity of the cost matrix calculation. Fisher vectors may help in reducing the dimensionality of the feature vector and also may eliminate the calculation of the complete cost matrix.

References

[1] Milford, M. J., and Wyeth G. F., SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, *IEEE International Conference on Robotics and Automation*, 1643-1649, 2012.

[2] Hafez, A. H., Agarwal, N. and Jawahar, C. V. Connecting Visual Experiences using Max-flow Network with Application to Visual Localization, *arXiv preprint arXiv:1808.00208*, 2018.

[3] Hafez, A. H., Arora, M., Krishna, K. M., and Jawahar, C. V., Learning multiple experiences useful visual features for active maps localization in crowded environments, *Advanced Robotics,* 30, 50-67, 2016.

[4] Hafez, A. H., Bhuvanagiri, S., Krishna, K. M., and Jawahar, C. V., On-line convex optimization based solution for mapping in VSLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4072-4077, 2008.

[5] Glover, A., Maddern, W., Warren, M., Reid, S., Milford, M. and Wyeth, G. F., OpenFABMAP: An open source toolbox for appearance-based loop closure detection, *IEEE International Conference on Robotics and Automation*, 4730-4735, 2012.

[6] Naseer, T., Burgard, W., and Stachniss, C., Robust visual localization across seasons, *IEEE Transactions on Robotics,* 34, 289-302, 2018.

[7] Arroyo, R., Alcantarilla, P.F., Bergasa, L.M., Yebes, J.J. and Gámez, S., Bidirectional loop closure detection on panoramas for visual navigation, *IEEE Intelligent Vehicles Symposium Proceedings*, 1378-1383, 2014.

[8] Garcia-Fidalgo, E. and Ortiz, A., Methods for Appearance-based Loop Closure Detection, *Springer International Publishing*, 1st Edition, 2018.

[9] Hafez, A. H., Singh, M., Krishna, K.M. and Jawahar, C.V., Visual localization in highly crowded urban environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2778-2783, 2013.

[10] Hafez, A.H., Arora, M., Krishna, K.M. and Jawahar, C.V., Learning multiple experiences useful visual features for active maps localization in crowded environments, *Advanced Robotics,* 30, 50-67, 2016.

[11] Murphy, L., Martin, S. and Corke, P., Creating and using probabilistic costmaps from vehicle experience, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4689-4694, 2012.

[12] Churchill, W. and Newman, P., Experience-based navigation for long-term localization, *The International Journal of Robotics Research,* 32,1645-1661, 2013.

[13] Dymczyk, M., Lynen, S., Cieslewski, T., Bosse, M., Siegwart, R. and Furgale, P., The gist of maps-summarizing experience for lifelong localization, *IEEE International Conference on Robotics and Automation (ICRA)*, 2767-2773, 2015.

[14] Labbé, M. and Michaud, F., Memory management for real-time appearance-based loop closure detection, *IEEE/RSJ international conference on intelligent robots and systems*, 1271-1276, 2011.

[15] Labbe, M. and Michaud, F., Appearance-based loop closure detection for online large-scale and long-term operation, *IEEE Transactions on Robotics,* 29, 734-745, 2013.

[16] Cummins, M. and Newman, P., FAB-MAP: Probabilistic localization and mapping in the space of appearance, *The International Journal of Robotics Research,* 27, 647-665, 2008.

[17] Schindler, G., Brown, M. and Szeliski, R., City-scale location recognition, *IEEE Conference on Computer Vision and Pattern Recognition*, 1-7, 2007.

[18] Dymczyk, M., Lynen, S., Cieslewski, T., Bosse, M., Siegwart, R. and Furgale, P., The gist of maps-summarizing experience for lifelong localization, *IEEE International Conference on Robotics and Automation (ICRA)*, 2767-2773, 2015.

[19] Zeng, F., Jacobson, A., Smith, D., Boswell, N., Peynot, T. and Milford, M., Enhancing underground visual place recognition with shannon entropy saliency, *Proceedings of the Australasian Conference on Robotics and Automation 2017*, 1-10, 2017.

[20] Abbas M Ali and Tarik A Rashid., Place recognition using kernel visual keyword descriptors, *SAI Intelligent Systems Conference (IntelliSys)*, 921–926, 2015.

[21] Qiao, Y., Cappelle, C. and Ruichek, Y., Place recognition based visual localization using LBP feature and SVM, *Mexican International Conference on Artificial Intelligence*, 393-404, 2015.

[22] Bellman, R., Some applications of the theory of dynamic programming—a review, *Journal of the Operations Research Society of America*, 2, 275–288, 1954.

[23] Myers, C., Rabiner, L. and Rosenberg, A., Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, *IEEE Transactions on Acoustics*, Speech, and Signal Processing, 28, 623–635, 1980.

[24] Sakoe, H. and Chiba, S., Dynamic programming algorithm optimization for spoken word recognition, *IEEE transactions on acoustics*, speech, and signal processing, 26, 43–49, 1978.

[25] Efrat, A., Fan, Q. and Venkatasubramanian, S., Curve matching, time warping, and light fields: New algorithms for computing similarity between curves, *Journal of Mathematical Imaging and Vision*, 27, 203–216, 2007.

[26] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara., The state of the art in online handwriting recognition, *IEEE Transactions on pattern analysis and machine intelligence*, 12, 787–808, 1990.

[27] Kuzmanic, A. and Zanchi, V., Hand shape classification using dtw and lcss as similarity measures for vision-based gesture recognition system, *EUROCON 2007-The International Conference on Computer as a Tool*, 264–269, 2007.

[28] Niennattrakul, V. and Ratanamahatana, C.A., On clustering multimedia time series data using k-means and dynamic time warping, *International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, 733–738, 2007.

[29] Gu, J. and Jin, X., A simple approximation for dynamic time warping search in large time series database, *International Conference on Intelligent Data Engineering and Automated Learning*, 841– 848, 2006.

[30] Müller, M., Dtw-based motion comparison and retrieval. *Information Retrieval for Music and Motion*, 211–226, 2007.

[31] Zhang, Z., Huang, K. and Tan, T., Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes, *18th International Conference on Pattern Recognition (ICPR'06)*, 3, 1135-1138, 2006.

[32] Vial, J., Noçairi, H., Sassiat, P., Mallipatu, S., Cognon, G., Thiébaut, D., Teillet, B. and Rutledge, D.N., Combination of dynamic time warping and multivariate analysis for the

comparison of comprehensive two-dimensional gas chromatograms: application to plant extracts, *Journal of Chromatography A*, 1216, 2866–2872, 2009.
[33] Müller, M., Mattes, H. and Kurth, F., An efficient multiscale approach to audio synchronization, *In ISMIR*, 546, 192–197, 2006.
[34] Kate, R.J., Using dynamic time warping distances as features for improved time series classification, *Data Mining and Knowledge Discovery*, 30, 283–312, 2016.
[35] Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y. and Keogh, E., Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm, *Knowledge and Information Systems*, 47, 1–26, 2016.
[36] Glover, A.J., Maddern, W.P., Milford, M.J. and Wyeth, G.F., FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day, *IEEE international conference on robotics and automation*, 3507-3512, 2010.
[37] Olid, D., Fácil, J.M. and Civera, J., Single-view place recognition under seasonal changes, *PPNIV Workshop at IROS 2018*, 2018.