Robotica: (2021) volume 39, pp. 557–571. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000570

Development of a Method for Data Dimensionality Reduction in Loop Closure Detection: An Incremental Approach

Leandro A. S. Moreira†‡*, Claudia M. Justel‡, Jauvane C. de Oliveira† and Paulo F. F. Rosa‡

†*Laboratório Nacional de Computação Científica, Brazil. E-mail: jauvane@acm.org*
‡*Instituto Militar de Engenharia, Brazil. E-mails: cjustel@ime.eb.br, rpaulo@ime.eb.br*

(Accepted June 19, 2020. First published online: July 17, 2020)

SUMMARY
This article proposes a method for incremental data dimensionality reduction in loop closure detection for robotic autonomous navigation. The approach uses dominant eigenvector concept for: (a) spectral description of visual datasets and (b) representation in low dimension. Unlike most other papers on data dimensionality reduction (which is done in batch mode), our method combines a sliding window technique and coordinate transformation to achieve dimensionality reduction in incremental data. Experiments in both simulated and real scenarios were performed and the results are suitable.

KEYWORDS: SLAM; Loop closure detection; Incremental dimensionality reduction; Mobile robots; Robot localization.

1. Introduction

Autonomous navigation of a mobile robot involves analyzing large amounts of data from different types of sensors, such as optical, ultrasonic, and lasers. The robot must be able to take online decisions about environment map building, determination of its own location, data association process, and development of path planning. In robotics, often simultaneous localization and mapping (SLAM) techniques[1] are used as a navigation tool for a robot placed in unknown environment. In this context, sensors installed in a mobile robot extract points of interest from the surroundings and probabilistic models are built upon data association, allowing to evaluate robot pose (i.e., position and orientation) as well as the location of those points of interest in the map. However, SLAM techniques have intrinsic uncertainties related to both location of landmarks and robot pose. As time goes by, such uncertainties increase, until the data association guarantees that a previously mapped area is now being visited again by the robot.[2] When this occurs, it is said that a "loop closure" is solved. Loop closure detection is a difficult problem. It demands good data association strategies to deal with the uncertainties, already mentioned, inherent in probabilistic models of SLAM techniques. Some of the techniques, called visual SLAM, can combine probabilistic models and computer vision methods and have been developed in recent years to make data association more robust. During the navigation process, such techniques use analysis of images and visual landmarks collected by the robot to solve the loop closure problem. This is because the association between sets of landmarks can be employed to determine how much similar the images are. Methods based on characteristic points

* Corresponding author. E-mail: leandromoreira75@gmail.com

extraction using local descriptors, such as SIFT[3] and SURF,[4] are examples of approaches that can be used to achieve this solution.

This paper proposes an incremental data dimensionality reduction method, which is applied to a loop closure detection approach, based on spectral description of images. This process is incremental because every time a dimensionality reduction method is applied to a set of points in a sequence, a different representation in low-dimensional space is generated. To handle this inconsistency, we propose to find a transformation matrix that guarantees the representation of the points in a unique basis on the tridimensional space $\mathbb{R}^3$. And in order to illustrate the performance of the proposed method, simulated experiments as well as experiments with a real robot are shown. The remainder of this paper is organized as follows: Section 2 presents some related works about spectral graph theory in mobile robotics and loop closure detection problem; Section 3 summarizes some data dimensionality reduction methods, including Diffusion Maps[5] – employed in our work; Section 4 illustrates the use of Diffusion Maps to detect loop closures in batch mode; Section 5 shows an incremental version of the Diffusion Maps; Section 6 presents results of the incremental method applied to loop closure detection in both simulated and real environments; and Section 7 is devoted to conclusions and future works.

2. Related Work

This section summarizes different papers about spectral graph theory for autonomous navigation and similarity analysis techniques for loop closure detection related to our approach. Also, based on literature review, we justify our choice of SIFT and SURF descriptors in the context of autonomous navigation.

2.1. Spectral graph theory and mobile robotics

Many previous studies based on spectral graph theory have shown how eigenvalues and eigenvectors of a graph can be applied to produce data analysis in autonomous navigation context, jointly to SLAM techniques. Spectral properties of adjacency and Laplacian matrices of a graph were presented in a theoretical framework by Zavlanos et al.[6] to analyze mobile robot networks. Concepts such as algebraic connectivity of a graph (Fiedler value) are employed for connectivity control and maintaining communication links between robots (e.g., in multirobot flocking and formation control). Olson et al.[7] have studied relaxed optimization problems regarding outliers rejection in autonomous navigation as well as parameter estimation for data association in SLAM. In this context, an adjacency matrix $\mathbf{A}$ is constructed for each problem, which is solved by the dominant eigenvector $\mathbf{u}$ related to Rayleigh quotient. Valgren et al.[8] used graph-based spectral clustering algorithms to generate topological online mapping of large indoor and outdoor environments using only appearance data. In this case, the method is a combination of algorithms presented by Ng et al.[9] and Verma and Meila[10] and obtains k clusters using the components of the k largest eigenvectors that are solving a generalized eigenvalue problem. Instead, in our paper, we use dominant eigenvectors as representative vectors related to the images that are analyzed and as orthonormal basis to build a low-dimensional mapping whereby these vectors are represented. Data obtained by 2D laser sensing from the navigation in large environments were used by Blanco et al.[11] to build hybrid metric-topological maps in an offline graph partitioning approach. That work represents a recursive version of the classic technique of spectral partitioning of graphs named Normalized Cut, due to Shi and Malik.[12] Forster et al.[13] have extended that idea and developed an online recursive graph partitioning method from data of radiofrequency tags placed in large environments. In that paper, the received signal strength indicator between tags is used to build the corresponding adjacency matrix of the graph in question. In our work, we apply a Gaussian function in two variables (dominant eigenvectors from images) to calculate the similarity among these images and to build the adjacency matrix. Yairi[14] presented a technique for 2D reconstruction of the environment and for obtaining the navigation map for mobile robot without the need for self-location. This approach is based on different graph-based spectral methods of machine learning and dimensionality reduction, such as the ISOMAP,[15] the Locally Linear Embedding (LLE),[16] and the Laplacian Eigenmaps.[17] In our paper, we also employ an spectral method to obtain data dimensionality reduction, but our main goal is to detect loop closure – a data association problem related to localization.

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

2.2. Visual analysis techniques and loop closure

Spectral decomposition of matrices and low-rank approximations were employed by Newman et al.[18] to present a similarity analysis technique which filters common visual information in SLAM, such as architectural patterns. However, this process is strongly dependent on building a vocabulary of visual words to make feasible loop closure detection. FAB-Map, a probabilistic method based on appearance, was presented by Cummins and Newman.[19] This method uses SURF descriptors[4] to build a BoW (Bag of Words) and is classified as an offline visual vocabulary approach. In our work, a single vector associated with these local descriptors is used without building any visual vocabulary. Cadena et al.[20] used stereo cameras combined to Conditional Random Fields to propose an offline visual vocabulary approach in order to improve local recognition process, when considering nearby and farway scenes. On the other hand, our method can use both stereo and monocular cameras. The binary descriptor BRIEF[21] and global descriptor *Gist*[22] were adapted by Sünderhauf and Protzel[23] to produce the BRIEF-Gist and develop a system to detect loop closure in large-scale scenarios with the construction of no vocabulary of visual words. But this approach is not capable to detect bidirectional loops. Arroyo et al.[24] employed global binary descriptors and Hamming distance to analyze panoramic images and identify bidirectional loop closures. Binary descriptors were used also by Garcia-Fidalgo and Ortiz[25] to construct a loop closure detection approach with no training phase. Such technique is based on discrete Bayes filter, and it builds a visual vocabulary in online mode. Our work does not depend on any training phase too, but only unidirectional loop closures are detected. It is important to note that none of these studies uses spectral methods to analyze the loop closure problem, such as it is done in ref. [26]. In that paper, the concept of dominant eigenvector is employed as an image spectral descriptor, and a data dimensionality reduction method[5] is used to detect loop closures in batch mode. In this paper, we extend the approach from ref. [26] to perform experiments in real and simulated environments with incremental method, through the combination of sliding window and coordinate transformation concepts.

2.3. How to choose a local descriptor?

There is no unanimity concerning how to choose the local descriptor that allows to reduce the execution time, as well as to increase the rate of precision in the recognition, of visual marks for a set of images. Mikolajczyk and Schmid[27] did a comparative analysis between SIFT – based on the precursor article from Lowe[3] – and other classes of descriptors, such as the steerable filters of Freeman and Adelson,[28] the differential invariants of Koenderink and Doom,[29] the complex filters of Schaffalitzky and Zisserman,[30] and the invariant moments of Gool et al..[31] Tests made in a set of 1000 modified images (rotated, with changes of scale, point of view and illumination) allow to conclude that, except for changes in illumination, the SIFT descriptors were more adequate to identify some correspondence between images. Gil et al.[32] have compared the behavior of different methods of detecting points of interest and descriptor formation under typical visual SLAM-based navigation conditions. In that comparison, the repeatability of points of interest and the invariance and distinction between the description methods were evaluated. This process was done for either images representing planar objects or images extracted from 3D scenarios. The repeatability of points of interest was analyzed in a set of images obtained from the same scene and taken from different distances, perspectives and lighting conditions – common situations in visual SLAM processes. On the other hand, the evaluation of the descriptors was done with a series of experiments to establish the correspondence between images. In addition to quality measurements of clustering, precision and recall measures were employed to decide which descriptor method was more appropriate. The tested methods of detecting points of interest were the Harris corners detector,[33] the Smallest Univalue Segment Assimilating Nucleus,[34] and the Maximally Stable Extremal Regions,[35] among others. SIFT and SURF were also tested as methods of producing descriptor vectors, as well as the Gradient Location-Orientation Histogram (GLOH).[27] Then, the authors concluded that GLOH and SURF were the most suitable methods for navigation with visual SLAM, and in case of camera rotation, SURF was recommended. Hartmann et al.[36] compared descriptors in the context of a graph-based visual SLAM technique. Both, the histogram descriptors – such as SIFT and SURF – as the so-called binary descriptors – such as BRIEF and BRISK (Binary Robust Invariant Scalable Keypoints), of Leutenegger et al.,[37] were compared . The ORB and FREAK (Fast Retina Keypoint) descriptors from Ortiz[38] were also used in the analysis. The objective was to determine the impact

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

of the choice of a descriptor formation method in terms of accuracy and speed in different scenarios. One of the scenarios tested by the authors was the RGB-D SLAM database, from Sturm et al..[39] Based on experimental results, and despite of the lower computational cost offered by the binary descriptors, the authors concluded that the method of detection and construction of SIFT descriptors is the best choice when considering robustness of the associations between images regarding visual SLAM methods.

3. Data Dimensionality Reduction

Huge amounts of high-dimensional data naturally arise in different fields of knowledge. Applications in several areas such as signal acquisition, image processing, classification and pattern recognition, and statistical learning are some examples. In such applications, what is intended is to make a simpler characterization of these datasets, generally through a meaningful representation in low dimension. One of the objectives of this characterization is to reveal the global and local structures of these sets. A property of any dimensionality reduction method should be stability, in the sense that the relationships between point-to-point distances in the original data space must be relatively preserved in the low-dimensional space. Over the past years, many methods have been developed, aiming at producing data dimensionality reduction. Some of these methods are briefly described in this section.

3.1. Principal Components Analysis

Principal components analysis (PCA)[40–42] is, at the best of our knowledge, the first and indeed the most popular method used to compute linear data dimensionality reduction in an unsupervised way. By finding a suitable linear transformation that maps the original data to a low-dimensional space, PCA constructs data representations where the variance in the data is maximal. Formally, consider a set of points $\mathbf{X}^{'} = \left\{\mathbf{x}_i^{'}, 1 \leq i \leq n\right\}$ in a high-dimensional space, where $\mathbf{x}_i^{'} \in \mathbb{R}^d$, $i = 1, ..., n$. The average between the points is $\mathbf{x}_m$. By centering $\mathbf{X}^{'}$ in $\mathbf{x}_m$, the matrix $\mathbf{X} = \{\mathbf{x}_i, 1 \leq i \leq n\}$, where $\mathbf{x}_i = \mathbf{x}_i^{'} - \mathbf{x}_m$, $i = 1, ..., n$, is obtained. The covariance matrix of $\mathbf{X}$ is given by symmetric and positive-semidefinite matrix $\Sigma = 1/n(\mathbf{X}\mathbf{X}^T)$, whose spectral decomposition is $\Sigma = \mathbf{U}\Lambda\mathbf{U}^T$. The transformation $\mathbf{Y} = \mathbf{U}^T\mathbf{X}$ gives us a reference system where $\mathbf{Y}$ has mean zero and a diagonal covariance matrix Λ, which contains the eigenvalues of Σ. The matrix $\mathbf{Y}$ has the vectors $\mathbf{y}_i$, which are obtained by rotating $\mathbf{X}$ according to a new orthogonal basis and axes which capture the maximum of the variance of $\mathbf{X}$, in descending order. In this new reference system, it is possible to discard variables with small variance. This means to project (with the best possible approximation) data from the original set $\mathbf{X}$ onto subspace spanned by the first p principal components, by means of $\mathbf{Y} = \mathbf{U}_p^T\mathbf{X}$, where $\mathbf{U}_p = \{\mathbf{u}_i, 1 \leq i \leq p\}$. This operation represents minor component truncation, used to achieve the reduction of an original dimension m for a reduced dimension p, with $p < m$. To do so, we select the top p from the m main components of a given dataset and ignore the others. This process preserves the variability of the original data, within the limits of this low-dimensional space.

3.2. Kernel PCA

Schölkopf et al.[43] has shown that it is possible to reformulate the traditional linear PCA to make nonlinear mappings. The variant method so-called Kernel PCA (KPCA) is constructed using a kernel function, similarly to other well-known techniques, such as SVM.[44] Given a mean-centered set of points $\mathbf{X} = \{\mathbf{x}_i, 1 \leq i \leq n\}$ and a positive-semidefinite kernel $\mathcal{K} : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$ (e.g., Gaussian function), KPCA method computes the symmetric and non-negative matrix $\mathbf{K} = [k_{ij}]_{n \times n}$, whose entries are represented by $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Instead to consider the covariance matrix $\Sigma = 1/n(\mathbf{X}\mathbf{X}^T)$, KPCA computes the principal eigenvectors $\mathbf{v}_i$ of the kernel matrix $\mathbf{K}$. Finally, to obtain the data representation in a low-dimensional space, the set $\mathbf{X}$ is projected onto the eigenvectors $\mathbf{a}_i$ of the covariance matrix, computed as $\mathbf{a}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{v}_i$, where λ_i are eigenvalues of $\mathbf{K}$.

3.3. Linear discriminant analysis

Consider a set $\mathbf{X} = \{\mathbf{x}_i, 1 \leq i \leq n\}$ with points $\mathbf{x}_i \in \mathbf{X}$ in a high-dimensional space. Suppose that $\mathbf{X}$ can be partitioned in m classes $\mathcal{C}_k$, such that $\mathbf{X} = \{\mathcal{C}_k, 1 \leq k \leq m\}$. Each class $\mathcal{C}_k$ has n_k points $\mathbf{x}_j^k \in \mathbb{R}^l$, $1 \leq j \leq n_k$. Thus, $n = \sum_{k=1}^{m} n_k$, and let be $\overline{\mathbf{x}^k} = \frac{1}{n_k}\sum_{j=1}^{n_k}\mathbf{x}_j^k$ and $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. The method *linear*

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

discriminant analysis (LDA)[45–47] is built to maximize between-class scattering and minimize within-class scattering, which are given respectively by the matrices $\mathbf{M}_b = \sum_{k=1}^{m} n_k \left(\overline{\mathbf{x}^k} - \overline{\mathbf{x}}\right)\left(\overline{\mathbf{x}^k} - \overline{\mathbf{x}}\right)^T$ and $\mathbf{M}_w = \sum_{k=1}^{m}\sum_{j=1}^{n_k}\left(\mathbf{x}_j^k - \overline{\mathbf{x}^k}\right)\left(\mathbf{x}_j^k - \overline{\mathbf{x}^k}\right)^T$. Maximization and minimization above can be written at the same time as $\arg\max_{\mathbf{U}} \frac{\left|\mathbf{U}^T\mathbf{M}_b\mathbf{U}\right|}{\left|\mathbf{U}^T\mathbf{M}_w\mathbf{U}\right|}$, and LDA seeks for the matrix $\mathbf{U}$ which solves this optimization problem. It is possible to show that the max-between-class and the min-within-class variabilities are found by the low-dimensional representation contained in the subspace spanned by the eigenvectors $\mathbf{u}$ corresponding to the $m-1$ largest eigenvalues from the equation $\mathbf{M}_b\mathbf{u} = \lambda\mathbf{M}_w\mathbf{u}$. The eigenvectors $\mathbf{u}$ are the columns of $\mathbf{U}$.

3.4. ISOMAP

ISOMAP[15] is a method that relates the points of a set $\mathbf{X} = \{\mathbf{x}_i,\ 1 \le i \le n\}$ in a high-dimensional space by constructing a neighborhood graph G, where every point $\mathbf{x}_i \in \mathbf{X}$ is connected to your k nearest neighbors $\mathbf{x}_{i_j} (j = 1, 2, ..., k)$. The shortest path between two points in the graph is an estimation for the geodesic distance between these points and can be computed by the Dijkstra algorithm.[48] The geodesic distances among all points in $\mathbf{X}$ are computed and stored in a matrix $\mathbf{D}$, and the representations $\mathbf{y}_i$ – in the low-dimensional space $\mathbf{Y}$ – corresponding to the points $\mathbf{x}_i$, are computed by minimizing the function $\phi(\mathbf{Y}) = \sum_{ij}(d_{ij}^2 - ||\mathbf{y}_i - \mathbf{y}_j||^2)$. It is possible to show that the minimum of the function $\phi(\mathbf{Y})$ is given by the spectral decomposition of the Gramian matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$. Thus, the principal eigenvectors of $\mathbf{K}$ are directly related to a low-dimensional representation of the original set of points $\mathbf{X}$.

3.5. Locally Linear Embedding

The method LLE[16] is similar to ISOMAP, concerning to construction of a graph to relate data in a high-dimensional space. But in turn, LLE seeks to accentuate local linearity characteristics by proposing to rewrite each $\mathbf{x}_i$ data point as a linear combination of its k nearest neighbors $\mathbf{x}_{i_j}$, thus getting a $\mathbf{w}_i$ vector of weights from the linear combination. This causes LLE to determine a hyperplane passing through the $\mathbf{x}_i$ point and its nearest neighbors. It can be shown that the representation of the original data $\mathbf{x}_i$ in a low-dimensional space $\mathbf{Y}$ is constructed by minimizing the function given by $\phi(\mathbf{Y}) = \sum_i ||\mathbf{y}_i - \sum_{j=1}^{k} w_{ij}\mathbf{y}_{i_j}||^2$. It is possible to show that the low-dimensional coordinates of the $\mathbf{y}_i$ representations that minimize the $\phi(\mathbf{Y})$ function can be obtained by calculating the eigenvectors corresponding to the smallest non-null eigenvalues of $(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$. In this inner-product, $\mathbf{I}$ is the identity matrix and $\mathbf{W}$ is the sparse matrix whose entries are 0 if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected by the neighborhood criterion and equals to w_{ij}, counterwise.

3.6. Laplacian Eigenmaps

Similarly to LLE, the method *Laplacian Eigenmaps*[17] also seeks for a low-dimensional representation from a neighborhood graph construction, and by preserving local properties of the connectivity relationships in the original data. Using the Gaussian function $w_{ij} = e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/\varepsilon}$, this method computes the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$, if they are connected by the neighborhood graph G. All the computed distances are stored in the sparse adjacency matrix $\mathbf{W}$. *Laplacian Eigenmaps* minimizes the function $\phi(\mathbf{Y}) = \sum_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2 w_{ij}$ to get the low-dimensional representations $\mathbf{y}_i$. By means of the degree matrix $\mathbf{M}$ – $\mathbf{M}$ is diagonal and $m_{ii} = \sum_j w_{ij}$ –, we obtain the Laplacian matrix $\mathbf{L} = \mathbf{M} - \mathbf{W}$. It is possible to show that $\phi(\mathbf{Y}) = \sum_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2 w_{ij} = 2\mathbf{Y}^T\mathbf{L}\mathbf{Y}$. Thus, it is easy to see that to minimize $\phi(\mathbf{Y})$ is equivalent to minimize the quadratic form $\mathbf{Y}^T\mathbf{L}\mathbf{Y}$. This optimization can be solved through the generalized eigenvalue problem $\mathbf{L}\mathbf{v} = \lambda\mathbf{M}\mathbf{v}$, where the d eigenvectors $\mathbf{v}_i$, corresponding to smaller non-null eigenvalues λ_i, define the low-dimensional representation $\mathbf{Y}$.

3.7. Diffusion Maps

Diffusion Maps is a data dimensionality reduction method introduced by Coifman and Lafon.[5] This method builds a coordinate system based on eigenvalues and eigenvectors in order to represent data in low dimension. Let $X = \{x_i\}_{i=1}^n$ be a finite set of points in a p-dimensional space. Consider that X

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

Algorithm 1 Basic version of Diffusion Maps

Input: $X = \{x_1, x_2, ..., x_n\}$, ε , s , t
Output: Low-dimensional mapping of X by eigenvalues and eigenvectors of $\mathbf{P}$
1. Build matrix $\mathbf{K}_{n\times n}$, $k_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\varepsilon}}$, $\forall i, j \in \{1, 2, ..., n\}$.
2. Obtain diagonal matrix $\mathbf{D}_{n\times n}$, $D_{ii} = \sum_{j=1}^{n} k_{ij}$.
3. Obtain the transition matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K}$.
4. Calculate eigenvalues $\lambda_1, ..., \lambda_s$ and eigenvectors $\psi_1, ..., \psi_s$ of $\mathbf{P}$.
5. Define the mapping $\Psi_t(x_i) = \left[\lambda_1^t\psi_1(x_i), \lambda_2^t\psi_2(x_i), \ldots, \lambda_s^t\psi_s(x_i)\right]$, $\forall i \in \{1, 2, ..., n\}$.

is on a differentiable manifold $\mho \subset \mathbb{R}^p$ not necessarily linear. One can see X as a graph $G = (X, E, w)$ since that w defines a measure of connectivity among their nodes x_i. This connectivity measure is called *kernel* and is represented by $\mathcal{K} : X \times X \to \mathbb{R}$. If two points x_i and x_j of X are connected by $\mathcal{K}$, then $k(x_i, x_j) = w_{i,j} \neq 0$. Finally, consider that $\mathcal{K}$ presents the symmetry and non-negativity properties. Usually, the so-called Gaussian kernel defined by $k(x_i, x_j) = e^{-\|x_i - x_j\|^2/\varepsilon}$ is considered. All evaluations of the kernel $\mathcal{K}$ can be stored in symmetric and non-negative matrix $\mathbf{K} = [k_{ij}]_{n\times n}$, so that $k_{ij} = k(x_i, x_j)$. Moreover, it is possible to submit $\mathbf{K}$ into a normalization process obtaining a matrix of transition probabilities $\mathbf{P}$ that represents a Markov chain on the state space X. For this, consider a diagonal matrix $\mathbf{D}_{n\times n}$ such that $\mathbf{D}_{ii} = \sum_j k_{ij}$. The matrix $\mathbf{P}$ is obtained from the matrix product $\mathbf{D}^{-1}\mathbf{K}$. Coifman and Lafon define, given a certain time t, a diffusion distance between points x_i and x_j denoted by $\mathcal{D}_t(x_i, x_j) = \|p_t(x_i, \bullet) - p_t(x_j, \bullet)\|_\xi$, where $\bullet$ indicates all points in X, $p_t(x_i, \bullet)$ is the transition probability from point x_i to points $\bullet$, and ξ is a certain weighting factor. The diffusion distance defined above establishes all distance relations $-$ weighted by ξ $-$ among all points in X. Diffusion Maps method defines through eigenvectors of $\mathbf{P}$ a diffusion coordinate system in Euclidean space $\mathbb{R}^s$ (with $s \ll p$) where points in X are mapped at the same time the intrinsic dataset structure is preserved. Thus, given t, an application $\Psi_t : X \to \mathbb{R}^s$ is built, obtaining an embedding of the dataset X in the Euclidian space $\mathbb{R}^s$ with the usual Euclidian norm. It can be shown that right eigenvalues of $\mathbf{P}$ form the sequence $1 = \lambda_0 > |\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_{n-1}|$ associated with right eigenvectors $\psi_0, \psi_1, ..., \psi_{n-1}$. Thus, given a point $x_i \in X$, its representation in space $\mathbb{R}^s$ on a certain instant t through diffusion coordinates will be as follows:

$$\Psi_t(x_i) = \left[\lambda_1^t\psi_1(x_i), \lambda_2^t\psi_2(x_i), \ldots, \lambda_s^t\psi_s(x_i)\right]$$

where $\psi_q(x_i)$ means the i-th component of q-th eigenvector of $\mathbf{P}$. Eigenvalue $\lambda_0 = 1$ and its correspondent eigenvector are not used in the construction of Ψ because all ψ_0 components are equal to each other. A pseudocode for Diffusion Maps is presented in Algorithm 1.

3.8. Discussion

Data dimensionality reduction methods based on graphs, such as ISOMAP proposed by Tenenbaum et al.,[15] LLE presented by Roweis and Saul[16] and Laplacian Eigenmaps by Belkin and Niyogi,[17] are particularly interesting. The establishment of some metric that relates the data in high dimension and yields a graph is the common idea in all those methods. The coordinates of the eigenvectors related to higher eigenvalues of a matrix associated with the graph are used to map the original data to low-dimensional spaces. This idea is also present in the Diffusion Maps method,[5] which in the context of autonomous navigation, was recently used by Chen et al.[49] to learn the geometry of a map with memory-efficient parameterization allowing to develop motion planning algorithms based on diffusion coordinates. In our work, we also apply Diffusion Maps, but in our case to detect loop closures through spectral description of visual datasets, in both batch and incremental modes. One of the reasons why we chose Diffusion Maps among graph-based methods is that it is the only one that uses, in addition to eigenvectors, eigenvalues for the data representation in lower dimensions. This

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

feature can produce better low-dimensional mappings, as pointed out by ref. [5]. Another reason is that none of the previous methods addresses, in its construction, the role of the density of the points that form the set X. Although we do not use this feature, Diffusion Maps, in its full version (see the family of diffusion distances concept[5]), allows to consider this information and can produce a mapping that shows the statistical characteristics of the sample, revealing something about the density of the input points.

4. Loop Closure Detection using Diffusion Maps

This section describes the loop closure detection with visual similarity analysis through the use of Diffusion Maps method. Given a set of images, each image is represented as a node of a graph in high dimension. This node is characterized by its dominant eigenvector, obtained by a matrix associated with the image, or by a matrix associated with local descriptors of the image. After this, the dimensionality reduction method is applied. And the evidences of loop closure will be determined. Basically, the applied methodology can be divided into three phases:

4.1. Phase A: Preprocessing

In the preprocessing phase, the dominant eigenvector $\mathbf{v_1}$ corresponding to a symmetric matrix should be obtained. This is done in one of two possible ways:

I – Dominant eigenvectors of the Gramian matrix of a pixels matrix Given an image, let $\mathbf{I}_{m\times n}$ be the corresponding matrix of pixels. Consider the square and symmetric Gramian matrix $\mathbf{M} = \mathbf{I}^T\mathbf{I}$ of size $n \times n$. $\mathbf{M}$ can be written, according to the spectral decomposition, $\mathbf{M} = \sum_{i=1}^{n} \mathbf{v_i}\mu_i\mathbf{v_i}^T$, where the μ_i are the eigenvalues of $\mathbf{M}$ and the column vectors $\mathbf{v_i}$ their respective orthonormal eigenvectors. It is possible to verify the existence of a significant spectral gap between μ_1 and the other eigenvalues. This is a property of Gramian matrices, as described by Johnstone.[50] Because of this gap, we use the rank one approximation $\mathbf{M_1} = \mathbf{v_1}\sqrt{\mu_1}\sqrt{\mu_1}\mathbf{v_1}^T$, where $\mathbf{v_1}^* = \sqrt{\mu_1}\mathbf{v_1}$ generates the outer product $\mathbf{M_1} = \mathbf{v_1}^* \bigotimes \mathbf{v_1}^*$. If two images are similar, so their respective $\mathbf{v_1}^* \in \mathbb{R}^n$ (or normalized version $\mathbf{v_1}$) have similar variational behavior. The dominant eigenvector $\mathbf{v_1}^i$ will be used as representative vector of a given image $\mathbf{I_i}$.

II – Dominant eigenvectors of local descriptors matrix Although it has higher computational cost, there is another preprocessing possibility, using the dominant eigenvector associated with the local descriptors matrix of an image. We use the square and symmetric matrix $\mathbf{M}_{r\times r} = \mathbb{D}^T\mathbb{D}$, where $\mathbb{D}_{k\times r}$ is the matrix of k local descriptors of r components obtained from an image $\mathbf{I}$. In this case, results were analyzed using SIFT[3] and SURF[4] descriptors, with $r = 128$ and $r = 64$, respectively. As in last section, the dominant eigenvector of matrix $\mathbf{M}_{r\times r}$, $v_1^i \in \mathbb{R}^r$, will be used as a representation of a given image $\mathbf{I_i}$.

4.2. Phase B: Data dimensionality reduction

We apply the Diffusion Maps method[5] in order to reduce the dimension of the representative vectors for the images obtained in *Phase A*. In the underlying graph of Diffusion Maps method, a node x_i – correspondent to one image $\mathbf{I_i}$ – will be represented by eigenvector $\mathbf{v_1}^i$, for all i. Thus, given the finite set V of eigenvectors $\mathbf{v_1}^i$ correspondent to a finite set of images, the Gaussian kernel $k(v_1^i, v_1^j) = e^{-\left\|v_1^i - v_1^j\right\|^2/\varepsilon}$ is defined and the mapping $\Psi_t : V \to \mathbb{R}^s$ is obtained, according to Section 3.7. After dimensionality reduction for $s = 3$, the low-dimensional representation in $\mathbb{R}^3$ is obtained, where the coordinate axes are represented by $\lambda_1\psi_1$, $\lambda_2\psi_2$ and $\lambda_3\psi_3$ – eigenvalues and eigenvectors of transition matrix $\mathbf{P}$ corresponding to graph. Note that here we set $t = 1$ for the mapping Ψ_t.

4.3. Phase C: loop closure in low dimension

The data dimensionality reduction process allows to represent each dominant eigenvector $\mathbf{v_1}$ as a point in a low-dimensional space. If two images are similar, then their dominant eigenvectors are also similar. Hence, computing the Euclidean distance between every pair of points in low dimension, a loop closure occurs when the points representing images in lower dimension are nearby. We considered the New College dataset,[19] a benchmark widely used by researchers in recognition and visual mapping, as an example of application of our method. This dataset is formed by pairs of images (left and right) of 640 × 480 pixels captured by a camera mounted on a pan-tilt support installed

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

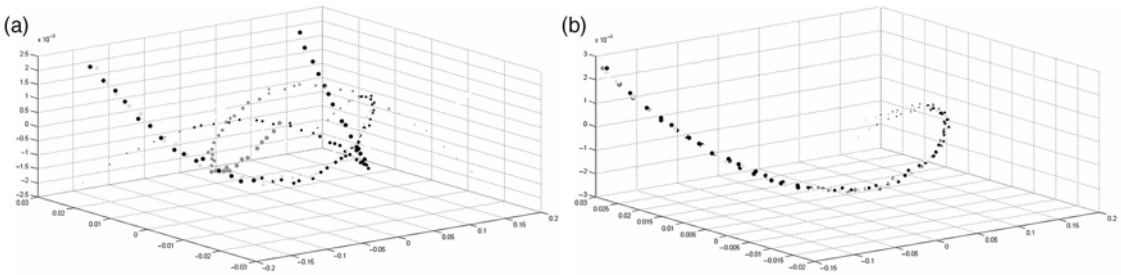

Fig. 1. Example of five instances of diffusion coordinates of the $k = 50$ initial points of a toroidal helix. In (a), the results obtained with representation in different orthonormal bases are shown. In (b), results after application of a homogeneous transformation matrix calculated for each instance.

on a robot. The images belong to New College campus, at the University of Oxford, and to apply our method, we have considered 992 pairs of them. The images were submitted to preprocessing approach described in Section 4.1. Each matrix of pixels $\mathbf{I}$ is the result of concatenating left and right images of each pair. As each image has 640×480 pixels, the corresponding matrix to a concatenated pair will be represented by $\mathbf{I}_{480\times1280}$, and the dominant eigenvector $\mathbf{v_1}$ of $\mathbf{M} = \mathbf{I}^T\mathbf{I}$ is calculated. Thus, each $\mathbf{v_1}^i$ used as input for the Diffusion Maps method has dimension 1280. The representation in low dimension ($s = 3$) obtained after applying Diffusion Maps to New College dataset, the matrix representation of detected loop closures by our method and the processing times for this dataset are shown in details in ref. [26].

5. Incremental Diffusion Maps

Typically, spectral data dimensionality reduction methods are used in batch mode, that is, all input data are known before the construction of a single global graph and the subsequent mapping in low dimension. However, in the context of mobile robotics, dimensionality reduction must be performed in an online way, simultaneously to navigation. So, it is necessary to adapt the structure of the proposed batch method to incremental processing. The generated graph and its diffusion coordinates representation might be updated while the robot senses the surroundings. As an illustration of the incremental problem adaptation, consider a sequence of points representing a discretization of a toroidal helix in $\mathbb{R}^3$. Every time the Diffusion Maps method is applied to a set of k points in a sequence, a representation by diffusion coordinates – not necessarily the same as the previous one – is generated. To handle this inconsistency, we propose to find a transformation matrix that guarantees the representation of the points by a unique basis on the tridimensional space $\mathbb{R}^3$. Figure 1 shows five computations of the diffusion coordinates in $\mathbb{R}^3$ of a subsequence with $k = 50$ points of the toroidal helix. In Fig. 1 part (a), the representations of the diffusion coordinates of the toroidal helix are initially on different bases of the three-dimensional space, while in part (b), these representations are in a unique basis. We achieve this representation after having computed homogeneous transformation matrices and applied them to the initial diffusion coordinates. A different shade of gray represents each one of the five computations. Details about this process are explained ahead in the course of this section.

Machine learning approaches are usually applied in graph-based methods for data dimensionality reduction with incremental processing that exist in the literature. In these cases, the training set is, in general, a sample of the full dataset to be analyzed. As an example, Lafon et al.[51] use a training set to infer the position in the low-dimensional mapping of a new sample of data, through an improved version of the Nyström extension.[52] Originally, the Nyström extension consists of finding numerical approximations for eigenvectors problems. Peng et al.[53] suggest an incremental version of the Laplacian Eigenmaps method.[17] This method seeks to optimize preservation of local neighborhood information from a new data point x_{n+1} introduced into a sample and, at the same time, correcting the adjacency relations of points affected by the introduced new point. These tasks are done in three steps. The first one consists of updating the adjacency matrix $\mathbf{W}$, which having initial size $n \times n$, is increased to $(n + 1) \times (n + 1)$, followed by the reconstruction of weights corresponding to connections among the points whose neighbors have been changed due to the insertion of x_{n+1}. In the second step, the low-dimensional representation y_{n+1} corresponding to x_{n+1} is computed. Finally,

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

a strategy is applied to find a hyperplane that passes through the points, whose neighborhoods have been changed and to compute the reconstruction weights, that are used to correct the low-dimensional representation of those points. Considering a Gaussian kernel defined on a training set Ω and the eigenvector ϕ_l of the kernel, it is possible to show that if a new point $\tilde{x}$ is included in Ω, then an approximation $\tilde{\phi}_l(\tilde{x})$ to the real $\phi_l(\tilde{x})$ can be obtained. More recently, Shmueli et al.[54] proposed an incremental version of Diffusion Maps to detect web traffic anomalies. This method uses the concepts of sliding windows and first-order approximations of eigenvalues and eigenvectors, according to ref. [55]. However, ref. [54] offers no strategy for the representation and correction of the coordinates of the points in low dimension. In this section, we present our incremental method that adapts the idea of sliding windows from ref. [54] and propose the use of homogeneous transformation matrices, in order to ensure consistency of the representation in low dimension.

The general idea of our incremental method is presented next. Given an initial sequence of images, we select a subsequence of k first images and compute one of the two preprocessings, presented in Section 4.1. As a result of the preprocessing, we obtain a set $V_{1,k}$ containing k dominant eigenvectors corresponding to each image. After that, we build, in batch mode, the mapping $\Psi_t : V_{1,k} \rightarrow \mathbb{R}^3$ that computes the low-dimensional representations of the elements in $V_{1,k}$. This low-dimensional representation defines a basis in $\mathbb{R}^3$ that will be the basis in which all the images might be represented. When a new image is collected by the sensors of the robot, its dominant eigenvector will be computed by the same preprocessing, and then a new subsequence of k images is considered, namely $V_{2,k+1} = (V_{1,k} - \mathbf{v_1}) \cup \mathbf{v_{k+1}}$. Now, the low-dimensional mapping $\Psi'_t : V_{2,k+1} \rightarrow \mathbb{R}^3$ is computed. Observe that $V_{1,k}$ and $V_{2,k+1}$ are two steps of the sliding window method, where the size of the window is k. And we can conclude that the representations in low dimension $\Psi_t(V_{1,k})$ and $\Psi'_t(V_{2,k+1})$ are different, because the correspondent eigenvalues and eigenvectors have been computed for different matrices. So, in order to obtain the same basis for all the low-dimensional representation of sets of k points, we consider that $\Psi_t(V_{1,k})$ is a system in which all the images representations will be computed. It is possible to consider that: (i) $\Psi_t(v_1)$ is a fixed point, and; (ii) there is a coordinate transformation that maps points in $\Psi'_t(V_{2,k+1})$ to points in $\Psi_t(V_{2,k+1})$. Then, we compute the coordinates of the vector $\Psi'_t(v_{k+1})$ using the same coordinates transformation. The sliding window and homogeneous transformation matrix method is repeated, as long as a new image is collected by the robot. More formally, let it be $\Psi_t(V_{1,k}) = [\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_k}]$, where $\mathbf{f_i} = \left[f_{i_x}\ f_{i_y}\ f_{i_z}\right]^T$, $\forall 1 \leq i \leq k$, the sequence of k initial points represented in low dimension through mapping $\Psi_t{:}V_{1,k} \rightarrow \mathbb{R}^3$. The diffusion coordinates Ψ_t define the coordinate system Σ_f. Also consider $\Psi'_t(V_{2,k+1}) = [\mathbf{m_2}, \mathbf{m_3}, ..., \mathbf{m_{k+1}}]$, where $\mathbf{m_i} = \left[m_{i_x}\ m_{i_y}\ m_{i_z}\right]^T$, $\forall 2 \leq i \leq k+1$, to be the sequence of k points represented through mapping $\Psi'_t : V_{2,k+1} \rightarrow \mathbb{R}^3$, where the domain of Ψ'_t was obtained by the advance of one step forward of a sliding window of size k into the sequence of images of the dataset. The diffusion coordinates Ψ'_t define the coordinate system Σ_m.

Naturally, coordinate systems Σ_f and Σ_m are defined by orthonormal bases in $\mathbb{R}^3$, but usually they are different from each other. However, it is possible to compute a homogeneous transformation matrix,[1] $\mathbf{H}_{m \rightarrow f}$, in such a way that the coordinates of the points in $\Psi'_t(V_{2,k})$ are mapped together and ordered in the coordinates of the points of $\Psi_t(V_{2,k})$, that is, $\mathbf{m_2} \rightarrow \mathbf{f_2}$, $\mathbf{m_3} \rightarrow \mathbf{f_3}$, ..., $\mathbf{m_k} \rightarrow \mathbf{f_k}$. So, Σ_m is transformed into Σ_f. The matrix $\mathbf{H}_{m \rightarrow f}$ − from here on denoted by $\mathbf{H}$ − is computed by Eq. (1)

$$\mathbf{H} \begin{bmatrix} \mathbf{m_2} & \mathbf{m_3} & \ldots & \mathbf{m_k} \\ 1 & 1 & \ldots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{f_2} & \mathbf{f_3} & \ldots & \mathbf{f_k} \\ 1 & 1 & \ldots & 1 \end{bmatrix} \tag{1}$$

It is straightforward to obtain, from Eq. (1), the overdetermined linear systems in Eqs. (2)–(4)

$$\left[\mathbf{M}\ 1\right] \left[H_{1,1}\ H_{1,2}\ H_{1,3}\ H_{1,4}\right]^T = \mathbf{F_x}^T \tag{2}$$

$$\left[\mathbf{M}\ 1\right] \left[H_{2,1}\ H_{2,2}\ H_{2,3}\ H_{2,4}\right]^T = \mathbf{F_y}^T \tag{3}$$

$$\left[\mathbf{M}\ 1\right] \left[H_{3,1}\ H_{3,2}\ H_{3,3}\ H_{3,4}\right]^T = \mathbf{F_z}^T \tag{4}$$

[1] $\mathbf{H}_{m \rightarrow f} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0} & 1 \end{bmatrix}$, where $\mathbf{R}_{3\times3}$ is a rotation matrix and $\mathbf{t}_{3\times1}$ is a translation vector.

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

Algorithm 2 Incremental version of Diffusion Maps with sliding window and coordinates transformation

Input: Initial sequence of k images, diffusion parameters ε and t
Output: Low-dimensional mapping of dominant eigenvectors V
1. Obtain the dominant eigenvectors $V_{1,k}$ (see Section 4.1).
2. Compute in batch mode in $\mathbb{R}^3$ the diffusion coordinates $[\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_k}] = \Psi_t(V_{1,k})$.
while new image $\mathbf{I}_{k+i}$ available ($i \geqslant 1$) **do**
1. Obtain the dominant eigenvector $\mathbf{v_{k+i}}$;
2. $V_{i+1,k+i} = (V_{i,i+k-1} - \mathbf{v_i}) \cup \mathbf{v_{k+i}}$;
3. Compute in batch mode in $\mathbb{R}^3$ the diffusion coordinates $[\mathbf{m_{i+1}}, \mathbf{m_{i+2}}, ..., \mathbf{m_{i+k-1}}] = \Psi_t(V_{i+1,k+i})$;
4. Compute the homogeneous transformation matrix $\mathbf{H}$ such that $\mathbf{H}\begin{bmatrix} \mathbf{m_{i+1}} & \mathbf{m_{i+2}} & \dots & \mathbf{m_{i+k-1}} \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{f_{i+1}} & \mathbf{f_{i+2}} & \dots & \mathbf{f_{i+k-1}} \\ 1 & 1 & \dots & 1 \end{bmatrix}$;
5. Update $\mathbf{f_{k+i}} = \mathbf{H}\mathbf{m_{k+i}}$;
6. Update $\Psi_t(V_{1,i+k}) \leftarrow \Psi_t(V_{1,i+k-1}) \cup \mathbf{f_{k+i}}$.
end while

where $\mathbf{M} = \begin{bmatrix}\mathbf{m_2}\ \mathbf{m_3}\ \dots\ \mathbf{m_k}\end{bmatrix}^T$, $\mathbf{F_x} = \begin{bmatrix}f_{2_x}\ f_{3_x}\ \dots f_{k_x}\end{bmatrix}$, $\mathbf{F_y} = \begin{bmatrix}f_{2_y}\ f_{3_y}\ \dots f_{k_y}\end{bmatrix}$, and $\mathbf{F_z} = \begin{bmatrix}f_{2_z}\ f_{3_z}\ \dots f_{k_z}\end{bmatrix}$.

Systems (2)–(4) can be easily solved by least squares method, and thus the 12 variables $H_{i,j}$, $1 \leq i \leq 3$ and $1 \leq j \leq 4$ are calculated. After that, the coordinate transformation $\mathbf{H}$ is applied to the point $\mathbf{m_{k+1}}$ (initially represented in the coordinate system Σ_m). Then, we obtain a good approximation $\mathbf{f_{k+1}}$ in Σ_f that will be included in $\Psi_t(V_{1,k})$. Thus, $\Psi_t(V_{1,k+1}) = \Psi_t(V_{1,k}) \cup \mathbf{f_{k+1}}$ and the process described above is repeated, until all images in the sequence have been represented in low dimension in only one coordinate system Σ_f. Algorithm 2 presents the pseudocode for the incremental version of Diffusion Maps.

6. Experimental Results

In order to illustrate the proposed incremental method, we consider two sets of images captured both simulated environment, as well as in real indoor environment.

6.1. Dataset 1: Images from simulated environment in Gazebo

We created an environment in Gazebo simulator to generate a sequence of 378 images with 640 × 480 pixels, captured by a virtual mobile robot with a RGB-D embedded camera. The robot was positioned at one point and only rotational motion around its axis was performed during just over two laps. Thus, a sequence of images corresponding to two loop closures was produced. Figure 2 shows some of the images in this set.

Images from this dataset were subjected to the preprocessing described in Section 4.1, with the use of SIFT and SURF descriptors. Here, we have chosen the first 100 dominant eigenvectors related to local descriptors matrices as the initial sequence of data. Then, the incremental process begins and loop closure detection results are correctly obtained. Figure 3 shows the representation in low dimension of the dominant eigenvectors set related to SIFT (Fig. 3(b)) and SURF (Fig. 3(d)) descriptors matrices, respectively. The grayscale used in those representations (Fig. 3(a)) is related to the order of the images in the sequence captured by the robot. Also, the matrix representation of the 20 largest Euclidean distances, between every pair of points in low dimension, is presented in Fig. 3(c) and (e). The sequences of loop closure detection are represented by the secondary diagonals.

6.2. Dataset 2: Images from real environment

In order to obtain the second dataset, we have employed a Pioneer 3DX robot with a RGB-D embedded camera in an indoor environment to capture a sequence of 565 images with 640 × 480 pixels. The robot performed a rectangular path in this second experiment, generating a sequence of images correspondent to only one loop closure. Again, images from this dataset have been preprocessed

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

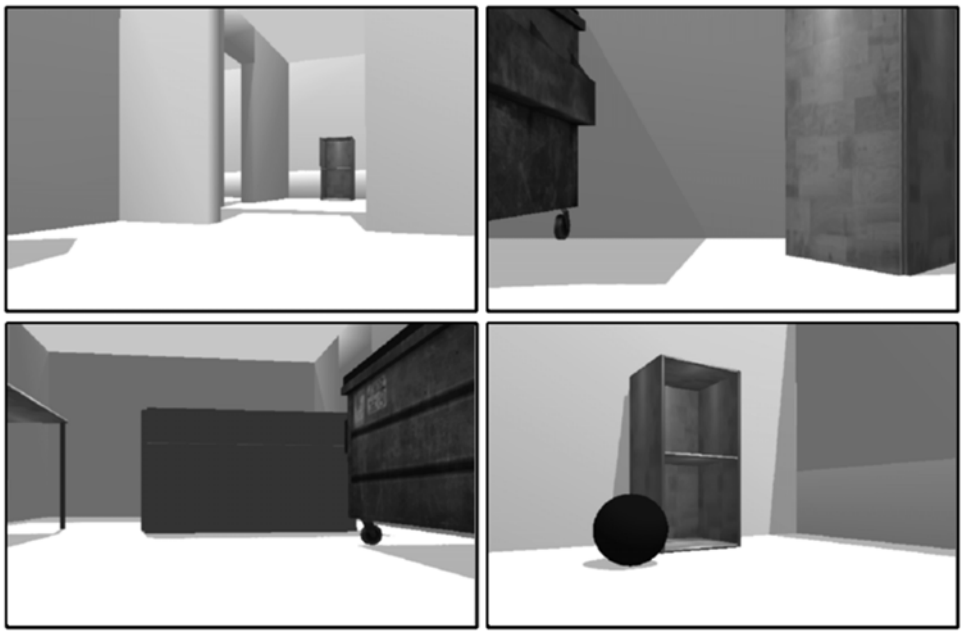

Fig. 2. Sample of dataset 1 captured in an environment created by Gazebo simulator.

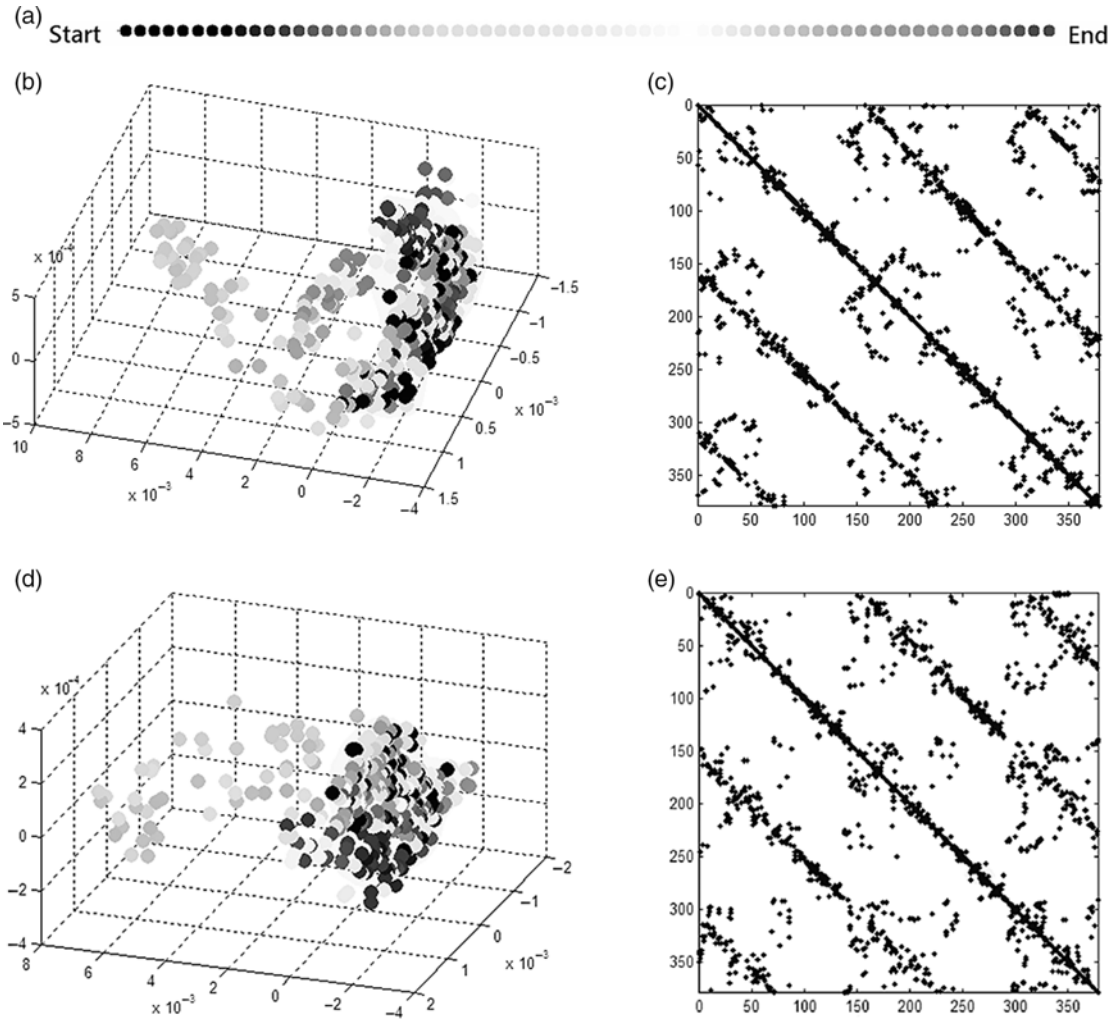

Fig. 3. Incremental Diffusion Maps: low-dimensional mappings ((b) and (d)), and Euclidean distances matrices ((c) and (e)) corresponding to dataset 1, using SIFT and SURF descriptors, respectively.

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

Fig. 4. Sample of dataset 2 captured in an indoor environment.

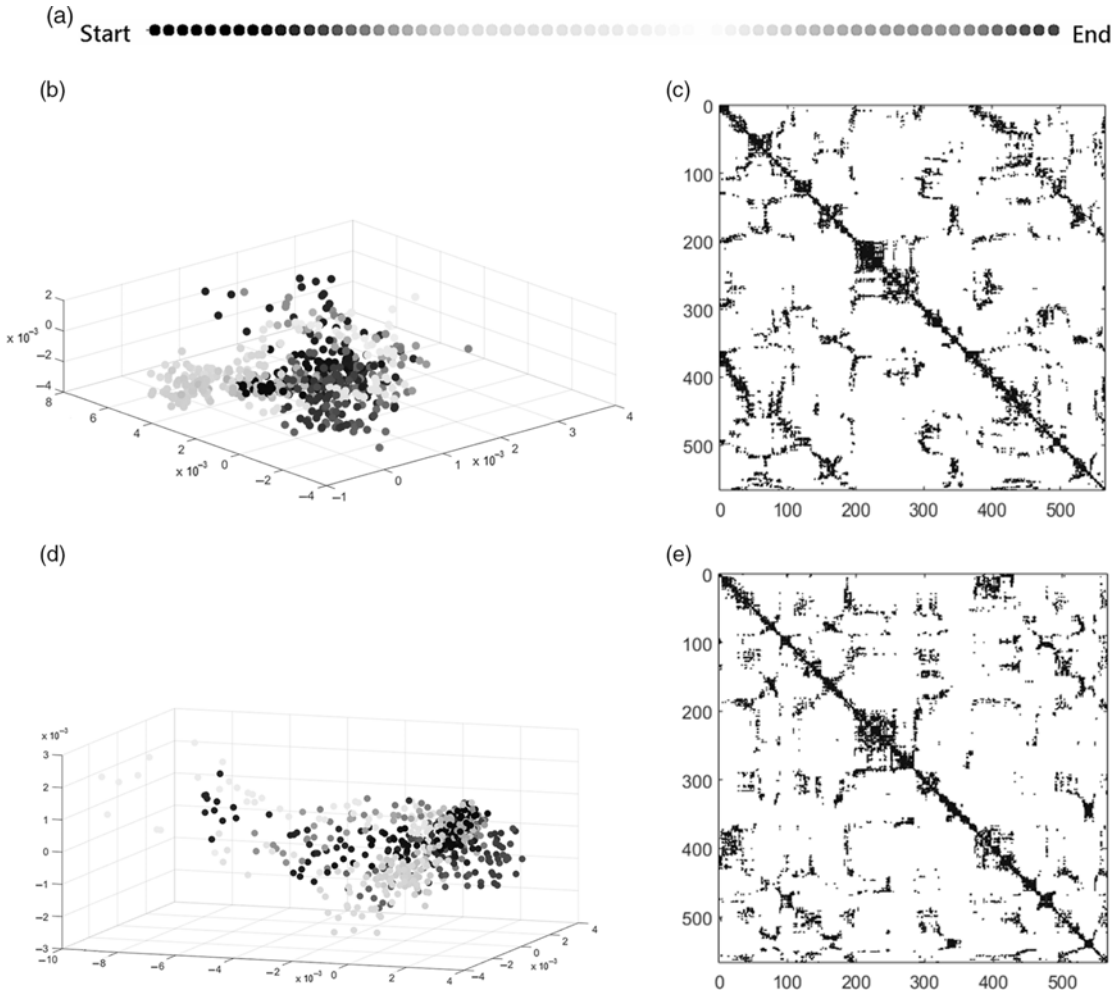

Fig. 5. Incremental Diffusion Maps: low-dimensional mappings ((b) and (d)), and Euclidean distances matrices ((c) and (e)) corresponding to dataset 2, using SIFT and SURF descriptors, respectively.

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

accordingly to Section 4.1, with the use of SIFT and SURF descriptors. Figure 4 shows some of the captured images. Here, we have chosen the first 200 dominant eigenvectors, related to local descriptors matrices, as the initial sequence of data. Then, the incremental process begins and loop closure detection results are correctly obtained again.

Figure 5 shows the representation in low dimension of the dominant eigenvectors set related to SIFT (Fig. 5(b)) and SURF (Fig. 5(d)) descriptors matrices, respectively. The grayscale used in those representations (Fig. 5(a)) is related to the sequence of images captured by the robot. Also, the matrix representation of the 20 largest Euclidean distances, between every pair of points in low dimension, is presented in Fig. 5(c) and (e). Thus, the incremental procedure, which combines the ideas of sliding window and change of coordinates by homogeneous transformation, can be directly applied in loop closure detection by visual analysis.

7. Conclusions

This paper proposed an approach for incremental data dimensionality reduction, based on spectral description of sets of images, to produce experimental results for loop closure detection in mobile robotics. We have combined a sliding window technique and coordinate transformation to create low-dimensional data representation in incremental mode (namely, as the robot senses its surroundings, the coordinates system of the lower dimension is updated). Experiments were performed in both computationally simulated and real environments, using Gazebo and a robot Pioneer 3DX, respectively. Results have shown that the proposed technique can detect loop closures correctly in situations where data collection is continuously performed, and the representation of the data computed in low dimension (with the consequent detection of loop closures) must be done online and simultaneously. An interesting work in the future will be to analyze the robustness of the method in relation to detection of false positives and false negatives. Precision and recall measures can be employed in this analysis.

Acknowledgments

The authors gratefully acknowledge the financial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001, and Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brazil (CNPq) – Grant No: 465586/2014-7.

Conflict of Interest

I and my co-authors confirm that this work is original and has not been published or is under consideration anywhere else. The authors declare that they have no relevant or material financial interests that relate to the research described in this paper. All authors have seen the paper and agree to publish it here.

References

1. J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous Map Building and Localization for an Autonomous Mobile Robot", *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91, IEEE/RSJ International Workshop* (1991) 1442–1447.
2. S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics* (Massachusetts Institute of Technology Press, Cambridge, MA, USA, 2005).
3. D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
4. H. Bay and L. Van Gool, "Surf: Speeded Up Robust Features", *Proceedings of 9th European Conference on Computer Vision* (2006) pp. 404–417.
5. R. Coifman and S. Lafon, "Diffusion maps", *Appl. Comput. Harmonic Anal.* **21**(1), 5–30 (2006).
6. M. M. Zavlanos, M. B. Egerstedt and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks", *Proc. IEEE* **99**(9), 1525–1540 (2011).
7. E. Olson, M. Walter, J. Leonard and S. Teller, "Single-Cluster Graph Partitioning for Robotics Applications", *Proceedings of Robotics Science and Systems* (2005) pp. 265–272 .
8. C. Valgren, T, Duckett and A. Lilienthal, "Incremental Spectral Clustering and its Application to Topological Mapping", *Proceedings of IEEE International Conference on Robotics and Automation* (2007) pp. 4283–4288.
9. A. Ng, M. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm", *Proceedings of Advances in Neural Information Processing Systems* (2001) pp. 849–856.
10. D. Verma and M. Meila, "A Comparison of Spectral Clustering Algorithms", University of Washington, Tech. Rep. UW-CSE-03-05-01 (2003).

11. J. L. Blanco, J. Gonzales and J. A. Fernández-Madrigal, "Consistent Observation Grouping for Generating Metric-Topological Maps that Improves Robot Localization", *Proceedings of International Conference on Robotics and Automation* (2006) pp. 818–823.
12. J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
13. C. Forster, D. Sabatta, R. Siegwart and D. Scaramuzza, "RFID-Based Hybrid Metric-Topological SLAM for GPS-Denied Environments", *Proceedings of International Conference on Robotics and Automation* (2013) pp. 5228–5234.
14. T. Yairi, "Map Building without Localization by Dimensionality Reduction Techniques", *Proceedings of the 24th International Conference on Machine Learning* (2007) pp. 1071–1078 .
15. J. Tenenbaum, V. Silva and J. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science* **290**(5500), 2319–2323 (2000).
16. S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding", *Science* **290**(5500), 2323–2326 (2000).
17. M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation", *Neural Comput.* **15**(6), 1373–1396 (2003).
18. P. M. Newman, D. M. Cole and K. L. Ho, "Outdoor SLAM Using Visual Appearance and Laser Ranging", *IEEE International Conference on Robotics and Automation (ICRA)* (2006) pp. 1180–1187.
19. M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance", *Int. J. Robot. Res.* **27**(6), 647–665 (2008).
20. C. Cadena, D. Gálvez-López, F. Ramos and J. D. Tardós, "Robust Place Recognition with Stereo Cameras", *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2010) pp. 5182–5189 .
21. M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary Robust Independent Elementary Features", *Proceedings of 11th European Conference on Computer Vision* (2010) pp. 778–792.
22. A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope", *Int. J. Comput. Vis.* **42**(3), 145–175 (2001).
23. N. Sünderhauf and P. Protzel, "BRIEF-Gist Closing the Loop by Simple Means", *International Conference on Intelligent Robots and Systems (IROS)* (2011) pp. 1234–1241.
24. R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Y. Torres and S. Gamez, "Bidirectional Loop Closure Detection on Panoramas for Visual Navigation", *2014 IEEE Intelligent Vehicles Symposium Proceedings* (2014) pp. 1378–1383.
25. E. Garcia-Fidalgo and A. Ortiz, "On the Use of Binary Feature Descriptors for Loop Closure Detection", *2014 IEEE Emerging Technology and Factory Automation* (2014) pp. 1–8.
26. L. Moreira, C. Justel and P. Rosa, "Experimental Implementation of Loop Closure Detection Using Data Dimensionality Reduction by Spectral Method", *Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT)* (2017) pp. 797–802.
27. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005).
28. W. T. Freeman and E. H. Adelson, "The design and use of steerable filters", *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(9), 891–906 (1991).
29. J. J. Koenderink and A. J. van Doom, "Representation of Local Geometry in the Visual System", *Biological Cybernetics* **55**(6), 367–375 (1987).
30. F. Schaffalitzky and A. Zisserman, "Multi-view Matching for Unordered Image Sets", *Proceedings of the 7th European Conference on Computer Vision-Part I* (2002) pp. 414–431.
31. L. J. van Gool, Luc and T. Moons and D. Ungureanu, "Affine/Photometric Invariants for Planar Intensity Patterns", *Proceedings of the 4th European Conference on Computer Vision-Volume I, ECCV 96* (1996) pp. 642–651.
32. A. Gil, O. M. Mozos, M. Ballesta and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM", *Mach. Vis. Appl.* **21**(6), 905–920 (2010).
33. C. Harris and M. Stephens, "A Combined Corner and Edge Detector", *Proceedings of Fourth Alvey Vision Conference* (1988) pp. 147–151.
34. S. Smith, "A New Class of Corner Finder", *Proceedings of the 3rd British Machine Vision Conference* (1992) pp. 139–148.
35. J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", *Proceedings of the British Machine Vision Conference* (2002) pp. 384–393.
36. J. Hartmann, J. Klussendorff and E. Maehle, "A Comparison of Feature Descriptors for Visual SLAM", *IEEE European Conference on Mobile Robots (ECMR)* (2013) pp. 56–61.
37. S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints", *Proceedings of the 2011 International Conference on Computer Vision* (2011) pp. 2548–2555.
38. R. Ortiz, "FREAK: Fast Retina Keypoint", *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012) pp. 510–517.
39. J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, D. Cremers, R. Siegwart and W. Burgard, "Towards a Benchmark for RGB-D SLAM Evaluation", *RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conference (RSS)* (2011).
40. K. Pearson, "On lines and planes of closest fit to systems of points in space", *Philos. Mag.* **2**(6), 559–572 (1901).

https://doi.org/10.1017/S0263574720000570 Published online by Cambridge University Press

41. H. Hotelling, "Analysis of a complex of statistical variables into principal components", *J. Edu. Psychol.* **24**(6), 417–441 (1933).
42. J. Jolliffe, *Principal Component Analysis* (Springer Verlag, New York, NY, USA, 1986).
43. B. Schölkopf, A. J. Smola and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Comput.* **10**(5), 1299–1319 (1998).
44. B. Boser, E. Bernhard, I. Guyon and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (1992) pp. 144–152.
45. K. Fukunaga, *Introduction to Statistical Pattern Recognition* (Academic Press, San Diego, CA, USA, 1990).
46. R. Duda, P. Hart and D. Stork, *Pattern Classification* (Wiley Interscience, New York, NY, USA, 2001).
47. Y. Shin and C. Park, "Analysis of Correlation Based Dimension Reduction Methods", *Int. J. Appl. Math. Comput. Sci.* **21**(3), 549–558 (2011).
48. E. Dijkstra, "A note on two problems in connexion with graphs", *Numer. Math.* **1**(1), 269–271 (1959).
49. Y. F. Chen, S. Y. Liu, M. Liu, J. Miller and J. P How, "Motion Planning with Diffusion Maps", *International Conference on Intelligent Robots and Systems (IROS)* (2011) pp. 1423–1430.
50. I. M. Johnstone, "On the distribution of the largest eigenvalue in principal components analysis", *Ann. Stat.* **29**(2), 295–327 (2001).
51. S. Lafon, Y. Keller and R. Coifman, "Data fusion and multicue data matching by diffusion maps", *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1784–1797 (2006).
52. C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines", *Neural Inf. Process. Syst.* **13**, 682–688 (2001).
53. P. Jia, J. Yin, X. Huang and D. Hu, "Incremental Laplacian eigenmaps by preserving adjacent information between data points", *J. Pattern Recognit. Lett.* **30**(16), 1457–1463 (2009).
54. Y. Shmueli, T. Sipola, G. Shabat and A. Averbuch, "Using Affinity Perturbations to Detect Web Traffic Anomalies", *Proceedings of the 10th International Conference on Sampling Theory and Applications (SampTA)* (2013) pp. 444–447.
55. Y. Shmueli, G. Wolf and A. Averbuch, "Updating kernel methods in spectral decomposition by affinity perturbations", *Linear Alg. Appl.* **437**(6), 1356–1365 (2012).