Sequential nonlinear manifold learning

S. Kumar, J. Guivant, B. Upcroft and H.F. Durrant-Whyte
Australian Centre for Field Robotics, University of Sydney, NSW 2006, Australia
Tel.: +612 93512186; Fax: +612 93517474; E-mail: suresh@acfr.usyd.edu.au

Received 5 December 2005
Revised 2 March 2006
Accepted 7 May 2006

Abstract. The computation of compact and meaningful representations of high dimensional sensor data has recently been addressed through the development of Nonlinear Dimensional Reduction (NLDR) algorithms. The numerical implementation of spectral NLDR techniques typically leads to a symmetric eigenvalue problem that is solved by traditional batch eigensolution algorithms. The application of such algorithms in real-time systems necessitates the development of sequential algorithms that perform feature extraction online. This paper presents an efficient online NLDR scheme, Sequential-Isomap, based on incremental singular value decomposition (SVD) and the Isomap method. Example simulations demonstrate the validity and significant potential of this technique in real-time applications such as autonomous systems.

Keywords: Feature representation, autonomous vehicles, machine learning

1. Introduction

Autonomous systems are typically equipped with multiple heterogenous sensors that return fundamentally different physical characteristics of the environment such as color, geometry, texture or reflectivity. The raw sensory data is set in a very high dimensional space that is not readily amenable to simple interpretation and reasoning tasks. The development of real time perception algorithms that generate compact and useful representations of unstructured dynamic worlds is critical to the development of next generation autonomous systems. Such algorithms have numerous potential applications ranging from data compression, robust data association to assist autonomous navigation, unsupervised feature selection to create terrain models and as a pre-processor to compact probability density estimation methods to assist multi-sensor data fusion.

A typical application of the proposed algorithm in a real-time autonomous system is shown in Fig. 1. Raw sensor data is processed to extract high dimensional features, that are subsequently compressed to serve as observations to a data fusion client [15]. Sequential-Isomap operates in the feature compression and sensor model update blocks of the framework in Fig. 1.

While traditional eigenvector methods such as Principal Component Analysis (PCA) and its numerous variants provide theoretically optimal representations from a data-compression standpoint, they are fundamentally unable to discover any usable nonlinear structure in the raw data. This limitation has motivated the development of various nonlinear embedding methodologies such as Kernel PCA [18], Isomap [19], Laplacian Eigenmaps [1] and Locally Linear Embedding (LLE) [17]. Most Nonlinear Dimensionality Reduction (NLDR) techniques presume that the data lies on or in the vicinity of a low

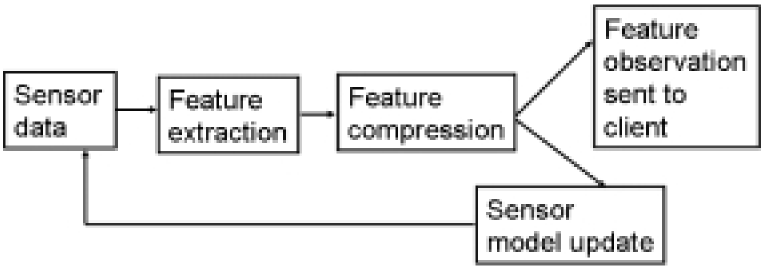

Fig. 1. Sensor data and representation flow chart in an autonomous system. The features extracted from raw sensor data are used to update the existing sensor model and are simultaneously available to clients for data fusion.

dimensional manifold and attempt to map the high dimensional data into a single low dimensional, global coordinate system.

Kernel PCA is based on the fact that PCA could be formulated entirely as dot products between data points, and hence the concepts of kernels and Reproducing Kernel Hilbert Spaces (RKHS) are effectively merged to develop a nonlinear version of PCA. The Isomap algorithm attempts to preserve the geodesic distances between faraway points (or the global structure) of the manifold through the solution of a dynamic programming problem and is quite successful at recovering highly nonlinear embeddings. The LLE technique recovers low dimensional embeddings that satisfy a basic premise that points that are close in the original high dimensional space remain so in the computed low dimensional embedding. The Laplacian Eigenmaps algorithm draws on the correspondence between the graph Laplacian, the Laplace-Beltrami operator on the manifold and connection to the diffusion equation to propose a geometrically motivated scheme to represent high dimensional data. In general, all these algorithms lead to the numerical solution of a large, symmetric and possibly sparse eigenvalue problem, and the eigenvectors associated with the lowest (or the highest) few eigenvalues form the basis of the computed embedding. All the cited spectral embedding methods have been cast in a unified framework in [2] as estimators of the eigenfunctions of a linear operator associated with a data dependent kernel.

As cast in their original form, all the NLDR approaches are inherently batch algorithms, which require that the entire data set be available for batch eigenvalue solution. Numerical algorithms that are used in the context of autonomous systems must be cast in a sequential form so as to work in real time as new observations pertaining to the state of an unstructured environment are acquired. Some typical examples [10] of such sequential algorithms cast in state space are the Kalman filter that is commonly used for real time tracking and the extended Kalman filter used for navigation. It would thus be extremely useful to have a sequential NLDR algorithms that could learn low dimensional representations from high dimensional sensory input in real time applications such as autonomous systems.

Sequential eigenvector extraction schemes have progressed significantly and most of them are based on a fundamental premise that the eigenvalues vary only gradually, so that iterative approaches can be used in real time. The simplest such approach has been described by Morita and Kanade [16] for the recovery of shape and motion from image streams. It involves using inverse iterations [11] in conjunction with guesses for the eigenvalues to rapidly converge to new eigenvector estimates. Goyal and Vetterli [12] have developed an innovative sequential algorithm for eigenvector computations. It uses stochastic descent which exploits the fact that a matrix can be diagonalized through a series of transformation matrices that are essentially Given's rotations, and the values of the rotations themselves are adapted online through a mean shift procedure.

While the iterative and descent based schemes appear promising, they suffer from a fundamental lack of robustness and almost always require problem specific tuning. Singular Value Decomposition (SVD) updating [11] on the other hand has received a lot of attention over almost two decades. The challenge here has been in dealing with the accumulation of numerical precision related errors over large updates that essentially destroy the orthogonality of the singular vectors. The work of Eisenstat and Gu [13] in this context is the culmination of this research into a robust, practical implementation, enabling large scale incremental updates and downdates. Brand [4] has recently implemented a similar incremental imputative SVD scheme in the context of web based recommender systems and structure from motion problems, that tackles added complexities due to measurement uncertainty and incomplete data. The incremental SVD scheme used in [4] is similar in principle to that used in this paper; however, the development of sequential NLDR algorithms based on incremental-SVD has not been addressed there.

The development of nonlinear dimensionality reduction methods and sequential singular value extraction schemes has occurred on essentially different tracks. There has been virtually no effort aimed at exploiting the rich developments in linear algebra within the framework of the NLDR algorithms so as to derive a whole class of unsupervised, *sequential* learning schemes. Useful out of sample extensions that predict low dimensional coordinates of test input from training data have been developed in [17] and [3], but these methods are not designed to adapt the pre-computed manifold to new input.

In the light of the arguments presented, a sequential NLDR scheme based on incremental SVD is developed here. The scheme fuses recent developments in SVD updating to a modified Isomap method to sequentially compute low rank representations of high dimensional data. The basic theory of the modified Isomap scheme is presented in Section 2. The theory and implementation of incremental SVD as presented in [13] is summarized in Section 4. Section 5 presents an incremental learning algorithm based on the concepts of the modified Isomap method and incremental SVD. Finally, example applications that demonstrate the capabilities of the incremental NLDR scheme are presented.

2. Theoretical aspects of the Isomap method

The Isomap method [19] formulates NLDR as the problem of finding a Euclidean feature space embedding of a set of observations that attempts to explicitly preserve their intrinsic metric structure; the metric structure is quantified as the geodesic distances between the points along the manifold.

The Isomap method starts out assuming that the data $\vec{X}$ lies on an unknown manifold embedded in the high dimensional observation space and attempts to reconstruct an implicit mapping $f : \vec{X} \rightarrow \vec{Y}$ that transforms the data to a low dimensional Euclidean feature space $\vec{Y}$, that optimally preserves the distances between the observations as measured along geodesic paths on the manifold. Significant steps in the Isomap algorithm are summarized next.

2.1. Nearest neighbor computation

Neighboring points on the manifold are determined based on the input space distances $d_X(i, j)$ between pairs of points $i, j \in \vec{X}$. Each input point is connected to adjacent points based either on the K nearest neighbors or all points within a fixed distance ϵ from the point under consideration. The neighborhood relations are expressed as a weighted graph G over the data points with edges of weight $d_X(i, j)$ between neighboring points.

2.2. Computation of geodesic distances

The length of a path in G is defined as the sum of the link weights along the path. The shortest path lengths d_G^{ij} between two nodes i and j in the graph G are computed through the Floyd's algorithm [9] that generally scales as $O(N^3)$ or the Dijkstra algorithm [6] that scales as $O(N^2 log(N))$, where N is the number of data points.

2.3. Graph embedding through Multi-Dimensional Scaling

Classical Multi-dimensional Scaling (MDS) is now used to compute a graph embedding in k dimensional space that closely respects the geodesic distances d_G^{ij} computed through the dynamic programming algorithms. The coordinate vectors $y_i \in \vec{Y}$ are chosen to minimize the cost function $E = \|\tau(d_G) - \tau(d_Y)\|_{L^2}$, where d_Y is the matrix of output space distances and the norm is the matrix L^2 norm $\sqrt{\sum_{i,j} (\tau(d_G) - \tau(d_Y))_{ij}^2}$, $\tau = \frac{1}{2} HSH$ is an operator that converts distances into inner products, $H_{ij} = \delta_{ij} - (1/N)$ is the centering matrix and $S_{ij} = (d_G^{ij})^2$ is the matrix of squared geodesic distances . The global minimum of the cost function is computed by setting the output space coordinates y_i to the top k eigenvectors of $\tau(d_G)$. Introducing a unit row vector $P = [111\ldots1]$ with N columns and an $N \times N$ translation matrix Q with constant entries $Q_{ij} = (\Sigma_{i=1}^N \Sigma_{j=1}^N S_{ij})$, the $N \times N$ matrix $\tau(d_G)$ is expressed as

$$\tau(d_G) = \frac{1}{2}\left(S - (\Sigma_{j=1}^N S_{ij})\frac{P}{N} - P^T \frac{(\Sigma_{i=1}^N S_{ij})}{N} + \frac{Q}{N^2}\right) \quad (1)$$

3. The modified Landmark Isomap method

The Landmark Isomap method was designed to overcome the significant computational burden involved in the Dijkstra algorithm and subsequent eigen-solution of a full symmetric matrix incurred in global Isomap. The theoretical description of a modified Landmark Isomap scheme used here closely adheres to the implementation provided by the original authors [20]. In this method, a small random subset n_L of the total number of data points N are designated as landmarks, and the distances of all the high dimensional points are evaluated only with respect to the landmarks. Thus, the distance matrix d_G^{ij} is only $N \times n_L$ as opposed to $N \times N$ in global Isomap. The cost of the Dijkstra algorithm correspondingly reduces to $O(n_L \times N \times log(N))$.

3.1. Computation of low dimensional embedding

Modified MDS is now applied to the rectangular distance matrix to compute a low dimensional embedding of the landmarks. The low dimensional embedding is obtained by computing the eigenvectors of the inner product matrix $B_n^T B_n$, where $B_n = -H_N \Delta H_n/2$, $H_N = \delta_{ij} - (1/N)$, $H_n = \delta_{ij} - (1/n_L)$ and Δ is a matrix of squared distances between all the samples and the landmarks. Introducing a unit row vector with n_L columns, $P^* = [11\ldots1]$ and an $N \times n_L$ translation matrix Q^* with constant entries $Q_{ij}^* = (\Sigma_{i=1}^N \Sigma_{j=1}^{n_L} \Delta_{ij})$, the $N \times n_L$ matrix B_n can be expressed as

$$B_n = \frac{1}{2}\left(\Delta - (\Sigma_{j=1}^{n_L} \Delta_{ij})\frac{P^*}{n_L} - P^T \frac{(\Sigma_{i=1}^N \Delta_{ij})}{N} + \frac{Q^*}{N \times n_L}\right) \quad (2)$$

This embedding in l dimensional space is designated as
$L = \left(\sqrt{(}\lambda_1)v_1^T, \sqrt{(}\lambda_2)v_2^T, \ldots, \sqrt{(}\lambda_l)v_l^T\right)^T$, where λ_i are the eigenvalues and v designate the eigenvectors.

3.2. Out of sample extensions

If $\triangle_n$ is the column-wise mean of Δ, a non-landmark point $\vec{y}$ can be embedded into the l dimensional space as

$$\vec{y} = (1/2)L^{\sharp}(\triangle_n - \triangle_y) \quad (3)$$

where $L^{\sharp}$ is the pseudo-inverse transpose of L and $\triangle_y$ is the vector of squared distances between the candidate point and the landmarks. Thus, the remaining $N - n_L$ non-landmark points can be embedded into the l dimensional space. Finally, a Principal Component Analysis (PCA) based transformation is applied to the embedded coordinates to realign the data with the coordinate axes.

The embedding computed by Landmark Isomap method is consistent with that computed by classical MDS at the landmark locations and is an estimate at the non-landmark locations. If the distance matrix between all the points and the landmarks can be represented exactly by a Euclidean configuration in R^l, and the landmarks are chosen such that their affine span in that configuration is l dimensional, the estimate at the non-landmark locations is accurate up to a rotation and translation.

3.3. Comments

Equation (3) provides a useful extension of the low dimensional coordinates [3] to non-landmark samples akin to mathematical induction. It is not designed to include new samples into the eigen-basis by incrementally adapting the pre-computed manifold.

A primary objective of this current work is to develop an algorithm that sequentially updates the low dimensional eigen-space and allows for a degree of adaptation of the manifold to fresh input. Such an algorithm is vital in the context of real time applications.

The Sequential-Isomap algorithm presented in this paper addresses these issues by explicitly reconstructing fully orthogonal singular vectors as each additional data point is input, and allows for the pre-computed embedding to reasonably adapt to new input.

4. Updating the singular value decomposition

The classical or Landmark versions of MDS lead to a symmetric eigenvalue problem that can be solved easily by traditional batch eigenvalue solvers. In the context of several real-time applications, notably autonomous systems, it is desirable to have an algorithm that incrementally updates the eigenvectors in response to sequential observations pertaining to an unknown state.

The problem of SVD updating has received a lot of attention in the linear algebra community over the last few decades, where the search for a fast and stable SVD updating scheme proved largely elusive, until the work of Eisenstat and Gu [13].

4.1. Preliminaries

The SVD of a matrix $A \in R^{m \times n}$ is defined as

$$A = U\Omega V^T \quad (4)$$

U and V are column-wise orthonormal and termed the left and right singular vectors and are identical for a symmetric matrix A. Ω is a diagonal matrix, whose entries are termed the singular values of A.

4.2. Connection to special symmetric eigenvalue problems

There are important relationships between the SVD of A and eigenvalue problems that may involve A [11]. Introducing a diagonal matrix $\Gamma = \Omega \times \Omega$, the SVD of $A^T A$ may be expressed as

$$A^T A = V^T \Omega U^T U \Omega V = V^T \Gamma V \tag{5}$$

due to the orthonormality of U or equivalently

$$(A^T A)V = \Gamma V \tag{6}$$

Thus the right singular vectors of A are identical to the eigenvectors of $A^T A$ and the eigenvalues of $A^T A$ are the squared singular values of A.

In the context of this paper, the right singular vectors of the $N \times n_L$ matrix B_n Eq. (2) are identical to the eigenvectors of $(B_n)^T B_n$ that are computed in the Landmark Isomap method.

4.3. SVD updating

In the context of a sequential implementation, A is repeatedly modified through the addition of rows and columns, and the goal is to be able to compute the singular values and singular vectors of the augmented version of A without resorting to the full batch solution. Thus, a tacit assumption that access to an existing SVD of A is available is made, and the focus is on the problem of updating this initial SVD. As the problem of adding a row to A is identical to the problem of adding a column to A^T, only the former problem is considered in this section.

The augmented matrix A' is written as

$$A' = \begin{pmatrix} A \\ a^T \end{pmatrix} \tag{7}$$

where $A' \in R^{(m+1)\times n}$ and a represents the added row. Considering first the case $m \geqslant n$, the left singular vector matrix and singular values can be partitioned as

$$U = \begin{pmatrix} U_1 \, U_2 \end{pmatrix}, \Omega = \begin{pmatrix} D \\ 0 \end{pmatrix} \tag{8}$$

where $U_1 \in R^{m\times n}$, $U_2 \in R^{m\times(m-n)}$ and $D \in R^{n\times n}$; this merely reflects the fact that there are at most n non-zero singular values in this case, and the left singular vectors are simply partitioned to show the correspondence.

It can easily be verified that A' can be rewritten as

$$A' = \begin{pmatrix} U_1 \, 0 \, U_2 \\ 0 \;\; 1 \;\; 0 \end{pmatrix} \begin{pmatrix} D \\ z^T \\ 0 \end{pmatrix} V^T \tag{9}$$

where $z = V^T a$ represents the projection of the added row onto the basis spanned by the right singular vectors. If the SVD of the broken arrowhead matrix $M = \begin{pmatrix} D \\ z^T \end{pmatrix}$ is given as $[Ww] \begin{pmatrix} \Omega \\ 0 \end{pmatrix} Q^T$, where $W \in R^{(n+1)\times n}$, $w \in R^{(n+1)}$ and $Q \in R^{n\times n}$, the SVD of the augmented matrix A' can be written as

$$A' = \begin{pmatrix} U_1' \, U_2' \end{pmatrix} \begin{pmatrix} \Omega \\ 0 \end{pmatrix} (VQ)^T \tag{10}$$

$U_1' = \begin{pmatrix} U_1 & 0 \\ 0 & 1 \end{pmatrix} W \in R^{(m+1)\times n}$, $U_2' = \left(w' \begin{pmatrix} U_2 \\ 0 \end{pmatrix} \right) \in R^{(m+1)\times(m+1-n)}$ and
$w' = \begin{pmatrix} U_1 & 0 \\ 0 & 1 \end{pmatrix} W.$

4.4. Connection to the SVD of the broken arrowhead matrix

The significance of the preceding equations in this subsection is the fact that the singular values and singular vectors of the augmented matrix A' can be computed from knowledge of the singular values and singular vectors of the broken arrowhead matrix M. The computation of the SVD of the broken arrowhead matrix itself will be detailed in the next subsection, where it is demonstrated that the computation generally scales as $O(n^2)$ where n is the number of singular values. Thus, for the matrix B_n Eq. (2), the computation scales as $O(n_L^2)$, as the number of singular values equals the number of chosen landmarks.

Once the SVD of the broken arrowhead matrix is computed, the SVD of A' is computed through the matrix products; i.e. the right singular vectors are updated as VQ, a computation that generally scales as $O(n^3)$ and which, in the case of a low rank matrix, reduces further to $O(r^3)$. Even if the matrix is not a low rank matrix, a significant contribution of [13] is that the computation could be performed in $O(n^2(log_2(\epsilon))^2)$, where ϵ is the machine precision, through novel use of the Fast Multipole Method first proposed in [5].

5. Discussion

In principle, the incremental SVD methodology could be used in conjunction with any of the eigenvector based NLDR algorithms in order to render them sequential; however the storage requirements of such sequential algorithms grow rapidly as U_1, U_2 and V that are full unsymmetric matrices in general need to be stored so as to compute the updated SVD of A'. In case of a low rank matrix, the number of columns of these matrices to be stored equal the rank of A', and hence the memory requirements are trivially small.

5.1. Computation of SVD of the broken arrowhead matrix

The broken arrowhead matrix M can be represented as $M = \begin{pmatrix} D \\ z^T \end{pmatrix}$
where $D = diag(d_1, d_2, d_3 \ldots d_n)$ and $z = [z_1, z_2, \ldots\ z_n]$. Assuming that the SVD of M is represented as

$$M = W \begin{pmatrix} \Omega \\ 0 \end{pmatrix} Q^T \tag{11}$$

and $\Omega = diag(\omega_1, \omega_2, \ldots \omega_n)$, the singular values obey the interlacing property [14]

$$0 < d_1 < \omega_1 < d_2 < \omega_2 < \ldots < \omega_n < d_n + \parallel z \parallel_2 \tag{12}$$

and the secular equation given by

$$F(\omega) = 1 + \Sigma_{i=1}^{n} \left(\frac{z_i^2}{d_i^2 - \omega^2} \right) = 0. \tag{13}$$

The left singular vectors $\mathbf{w}$ satisfy Eq. (16)

$$\mathbf{Fr}_1 = \big(d_1 z_1/(d_1^2 - \omega_j^2), \ldots, d_n z_n/(d_n^2 - \omega_j^2), -1\big)^T \tag{14}$$

$$Fr_2 = \sqrt{(1 + \Sigma_{i=1}^n (d_i^2 z_i^2/(d_i^2 - \omega^2)^2)} \tag{15}$$

$$w_j = \frac{\mathbf{Fr}_1}{Fr_2}, j = 1 \ldots n \tag{16}$$

$$w_{n+1} = \big(d_1/z_1, \ldots d_n/z_n, -1\big)^T / \sqrt{(1 + \Sigma_{i=1}^n (z_i^2/d_i^2))} \tag{17}$$

while the right singular vectors $\mathbf{q}$ satisfy

$$q_j = \frac{\big(z_1/(d_1^2 - \omega_j^2), \ldots z_n/(d_n^2 - \omega_n^2)\big)^T}{\sqrt{(\Sigma_{i=1}^n (z_i^2/(d_i^2 - \omega_j^2)^2))}}. \tag{18}$$

The stable computation of the roots of the secular equation through standard bisection based solvers so as to ensure accurate singular values and orthonormal singular vectors is further elaborated on in [13]. The key idea is a slight reformulation of the secular equation and the introduction of an appropriate stopping criterion, so that a valid solution of the singular values that satisfies the interlacing property results from the numerical scheme.

6. The Sequential-Isomap algorithm

6.1. Overview

The incremental SVD algorithm outlined in Section 4 sequentially computes approximations to the singular values and singular vectors within machine precision as new rows a are appended to augment the original matrix A. Thus, in principle, it is easy to render any NLDR algorithm sequential by applying the incremental SVD scheme as long as the sequential NLDR scheme can be formulated in terms of addition of rows and columns.

6.2. Offline computations

Sequential-Isomap assumes that an initial low dimensional embedding of a training set is available in terms of the initial left singular vectors U_0, the initial singular values W_0 and the right singular vectors V_0. In the context of the modified Landmark Isomap method, the following computations are performed offline.

1. A weighted graph over the training samples (Section 2) is computed using K nearest neighbors. An $N \times K$ distance matrix D^0 comprising of Euclidean distances between each sample and the K nearest neighbors is used in this implementation.
2. The *fixed* squared geodesic distances Δ between all training samples and the landmarks are computed through Dijkstra's shortest path algorithm.

3. An initial low dimensional embedding is computed through the SVD of the initial inner product matrix B_n^0.
4. The $1 \times n_L$ row vector $T_2 = \Sigma_{i=1}^{N} \Delta_{ij}$ and scalar $T_4 = \Sigma_{i=1}^{N} \Sigma_{j=1}^{n_L} \Delta_{ij}$ are saved for use in incremental computations.

6.3. *Online computations*

A new data point contributes an additional row to B_n^0 and hence the incremental SVD scheme could be used to compute the updated singular values and singular vectors. The key assumption in Sequential-Isomap is that the new data point does not alter any of the pre-computed geodesic distances between any existing samples and the landmarks. This assumption is essential for the formulation of a sequential manifold learning algorithm based on mere addition of rows or columns. The additional row a that augments B_n^0 is approximated by the sum of four terms essentially derived from Eq. (2). Introducing a $1 \times n_L$ row vector Δ_{new} that represents the squared shortest paths between a new sample and the landmarks, computed on the basis of the existing weighted graph, a is estimated as

$$a = \frac{1}{2}\left(\Delta_{new} - (\Sigma_{j=1}^{n_L} \Delta_{new})P^* - \frac{T_2 + \Delta_{new}}{N_t} + \frac{T_4 + \Sigma_{j=1}^{n_L} \Delta_{new}}{N_t n_L}\right) \tag{19}$$

where N_t is the current number of samples including the training samples. T_2 and T_4 are subsequently updated as

$$T_2 = T_2 + \Delta_{new} \tag{20}$$

$$T_4 = T_4 + \Sigma_{j=1}^{n_L} \Delta_{\text{new}}$$

Equation (19) provides an estimate of the new row that augments the inner product matrix. The reliability of this estimate depends strongly on the accuracy of the approximated geodesic distances. The study of the reliability of a for smaller training sets is open to research. In practice, an alternate estimate for a has been found effective especially for smaller training sets Eq. (21).

$$a = \frac{1}{2}\left(\Delta_{new} - (\Sigma_{j=1}^{n_L} \Delta_{new})P^* - \frac{T_2}{N} + \frac{T_4}{N \times n_L}\right) \tag{21}$$

The update step Eq. (20) is eliminated if this estimator is chosen.

6.4. *Significant steps in Sequential-Isomap*

1. An initial low dimensional embedding of the training data is computed offline in terms of the SVD of B_n^0. The distance matrix D^0, singular values W_0 and right singular vectors V_0 are saved for online use.
2. Every additional sample is treated as a non-landmark sample in this implementation. The Euclidean distances of this sample from the K nearest neighbors are computed, and subsequently input to Dijkstra's algorithm to compute geodesic distances between the new point and the landmarks. All existing nodes in the weighted graph are used in this computation. The key assumption here is that the new data point does not alter any of the pre-computed shortest paths.

3. The additional row a that augments B_n^0 is computed from the known geodesic distances Eq. (19). The projection of this row z onto the space spanned by the right singular vectors V_0 is computed as $V_0^T a$. The augmented matrix is referred to as B_n hereafter.
4. The SVD of the broken arrowhead matrix $(W_0 \ \ z)^T$ is computed through a bisection based solver with an appropriate stopping criterion as described in Section 4.
5. The SVD of the augmented matrix B_n is now computed according to Eq. (10). The matrix products in this case do not pose a storage and computational problem as the number of columns in V_0 or U_0 is effectively n_L, the number of chosen landmarks.
6. The new sample is appended to the weighted graph so that it may serve as a neighbor for data points acquired in future.
7. Finally, a PCA like transformation is performed as in the Landmark Isomap method to realign the embedded coordinates into a global set of axes. The dominant cost in this step is the multiplication of B_n $(N_t \times n_L)$ with the current singular vectors V $(n_L \times n_L)$. In a practical implementation, this step need only be performed periodically in non-critical autonomous missions. This will result in existing low dimensional estimates marginally lagging the current sample states.

6.5. *Possible extension*

In this implementation, new samples are always regarded as non-landmark samples. It is also possible to designate new samples as landmarks, and this would result in the formulation of a sequential algorithm based on the addition of a row and a column to B_n. The incremental SVD procedure developed in Section 4 must now be applied twice to incorporate the new row and column. It is thus important to develop robust heuristics to aid landmark selection in a real-time implementation. This remains an open area of research at this juncture.

6.6. *Limitations*

The fact that the pre-computed geodesic distances are not allowed to change is crucial to the development of an incremental algorithm. The direct consequence of this assumption is that the estimation of the manifold cannot significantly improve the manifold between the training points. However, the manifold may be expanded in new directions.

It is also important to realize that the best solution from Sequential-Isomap would coincide with a batch modified Landmark-Isomap computation using identical algorithmic parameters. The convergence of the incremental solution to the batch modified Landmark Isomap solution in terms of number of training points and number of neighbors is studied numerically in the examples that follow. Theoretical research to provide realistic bounds on the accuracy of the additional row a Eq. (19) remains open. The convergence of the modified Landmark Isomap method to the underlying manifold also remains open to theoretical research. In practice, valuable results have been obtained on realistic data as is evident from the examples presented in this paper.

6.7. *Computational costs and memory requirements*

The computational complexity and memory requirements (in double precision words) of each step in the online execution are tabulated here using notation introduced throughout the paper. The computational costs are dominated by the shortest path computations as the number of samples added to the weighted graph increases. The incremental SVD execution is very efficient as the execution costs are

Table 1
Computational costs and memory requirements of Sequential-Isomap

Step	Computational cost	Memory requirements	Description
2	O($K \times N_t \times log(N_t)$)	$(N_t \times K + N_t \times n_L)$	D^0, Δ
3	O(n_L^2)	$n_L \times 1$	New row a
4	O(n_L^2)	n_L	W_0, z
5	O(n_L^3)	n_L^2	V_0, Q
7	O($N_t n_L^2$)	$N_t \times n_L$	B_n

dependent on the number of chosen landmarks, which is significantly smaller than the number of current samples. It is to be noted that Step 1 in Sequential-Isomap is performed offline and has a computational complexity $O(Kn_L NlogN + n_L^2 N)$ [7]. Table 1 summarizes the computational complexity and memory requirements of Sequential-Isomap.

7. Terminology and application overview

7.1. Terminology

The performance of Sequential-Isomap is contrasted with batch and sequential induction methods in subsequent sections. The terminology used in these sections is summarized here.

1. Batch Isomap implies a full Isomap solution without use of landmarks (Section 2).
2. Batch Landmark Isomap implies a full Landmark Isomap solution that recomputes the weighted graph, applies landmark MDS, extends the embedding to include non-landmark samples and performs the PCA transformation.
3. Landmark Isomap Induction is similar to Sequential-Isomap except for steps 3–5. These steps are replaced by the direct application of Eq. (3) to estimate low dimensional estimates for non-landmark samples.

7.2. Application overview

The following sections numerically assess the validity of the low dimensional representations computed by Sequential-Isomap. The simplest test presented in the first example (Section 8) tests the ability of the online algorithm to replicate a known two dimensional manifold. The next example (Section 9) assesses the ability of the sequential algorithm to replicate known dimensions in a higher dimensional human face dataset. The subsequent examples apply the methodology to real visual data acquired from autonomous ground and underwater vehicles, and demonstrate the accuracy and applicability of Sequential-Isomap representations in real-time autonomous systems. The convergence characteristics of the solution are studied thoroughly with respect to varying training set sizes, and the computational complexity of the algorithm is contrasted with respect to the batch computations.

8. The S manifold – A 2 dimensional manifold embedded in 3 dimensions

An analytically generated three dimensional manifold that intrinsically has a two dimensional embedding is depicted in Fig. 2. The goal of any unsupervised learning algorithm, batch or sequential, is to automatically discover the global coordinates intrinsic to the two dimensional embedding without any explicit directives on how the data is to be mapped onto the low dimensional space.

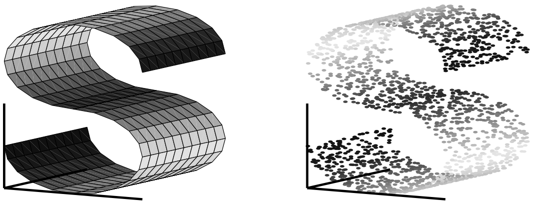

Fig. 2. Analytically generated and randomly sampled S-shaped nonlinear manifold. Left – High dimensional data. Right – Randomly generated high dimensional samples.

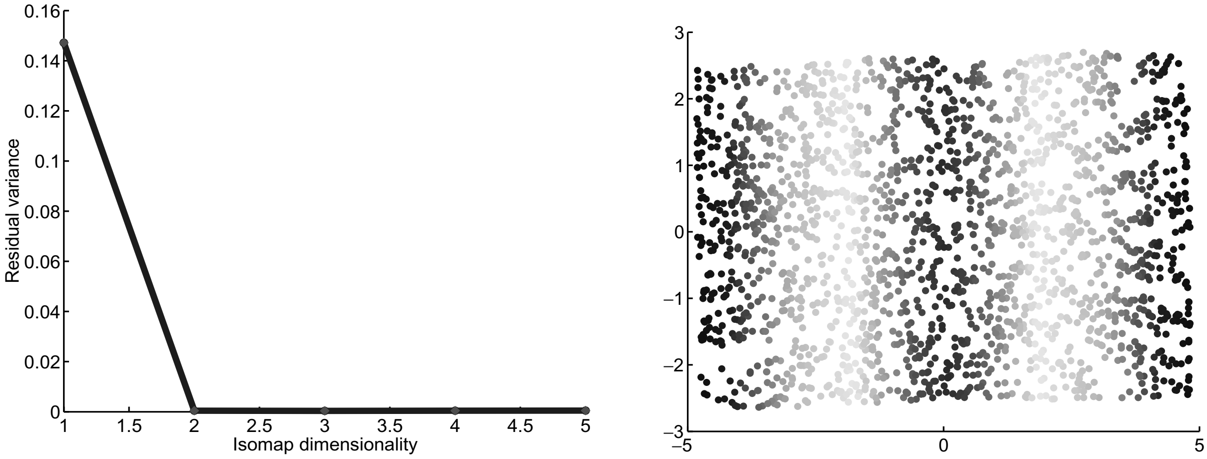

Fig. 3. Manifold dimensionality estimated by Isomap. (left) An examination of the residual variances indicates the two dimensional nature of the manifold. Isomap computed two dimensional embedding (right). High dimensional neighborhoods are preserved in two dimensions.

8.1. Batch Isomap embedding

The picture on the right in Fig. 3 shows a two dimensional embedding computed by batch Isomap that essentially unwraps the three dimensional manifold. Isomap also allows an estimation of manifold dimensionality through a plot of the residual variance with respect to the embedding dimensions. Figure 3 clearly shows the two dimensional nature of the raw three dimensional data through the vanishing residual variances. The sampled manifold consisted of two thousand random data points in all the cases presented.

8.2. Sequentially computed embeddings

Sequential-Isomap was used to compute low dimensional embeddings of the S-curve. The size of the training set was varied from 200 to 400 randomly selected points. Fifty points were randomly selected from the training set to serve as landmarks. The remainder of the data points were sequentially input to Sequential-Isomap.

Figure 4 shows the computed embeddings for various choices of training set sizes (N) and neighborhoods (K). For small sizes of the initial training set ($N = 200$), Sequential-Isomap embeddings for $K = 8$ are most representative of the batch solution. The solutions computed with $N = 400$ and $K = 10 - 12$ seem optimal in terms of preservation of neighborhoods and geometric structure. The aspect ratio (1 in 2) of the batch embedding (Fig. 3) is reasonably well preserved in all the cases cited here.

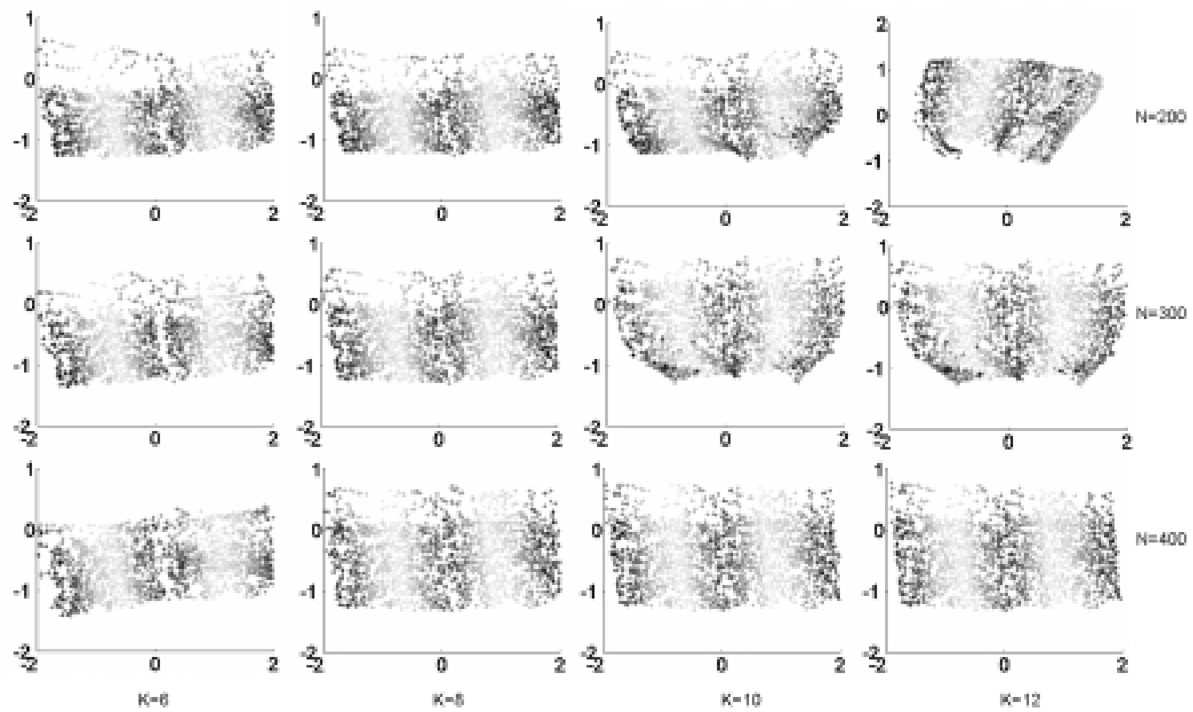

Fig. 4. Variation of the geometry of the computed Isomap embedding with varying training set size and number of neighbours. The quality of the embedding improves with an increasing set size and an optimal number of neighbours.

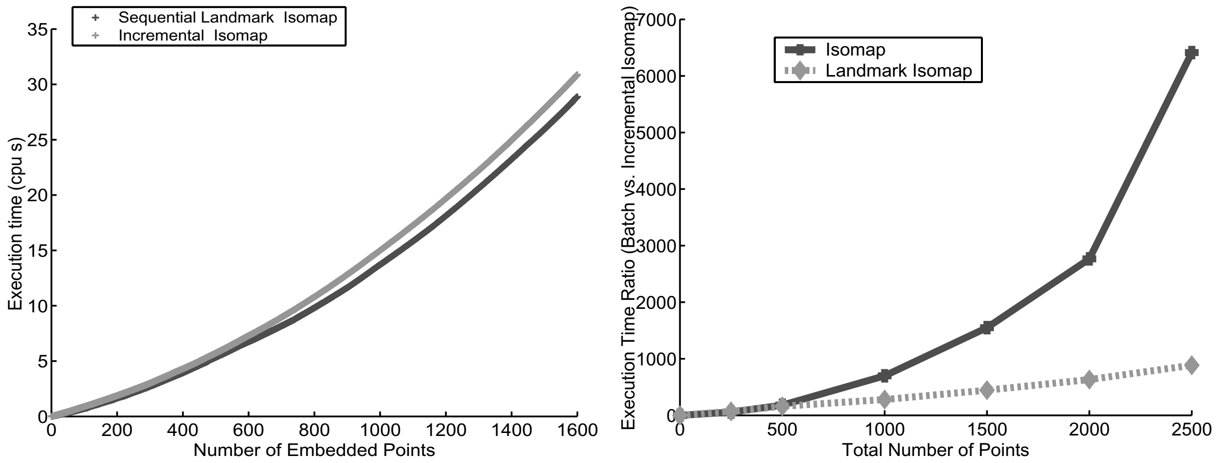

Fig. 5. Execution times of Sequential-Isomap and direct application of the Landmark Isomap formula (left). Sequential-Isomap is marginally more expensive due to the added cost of incremental SVD. Execution time ratios of batch algorithms (Isomap, Landmark Isomap) and Sequential-Isomap (right). The PCA update step was performed 5 times during the course of this simulation. The maximum time consumed by this step was 0.9 cpu s. Sequential-Isomap is observed to be increasingly competitive with growth in the number of samples.

8.3. Computational cost considerations

Figure 5 (*left*) compares the execution times of Sequential-Isomap and a direct application of Eq. (3). The addition of incoming samples to the graph structure incurs added costs in Dijkstra's algorithm in

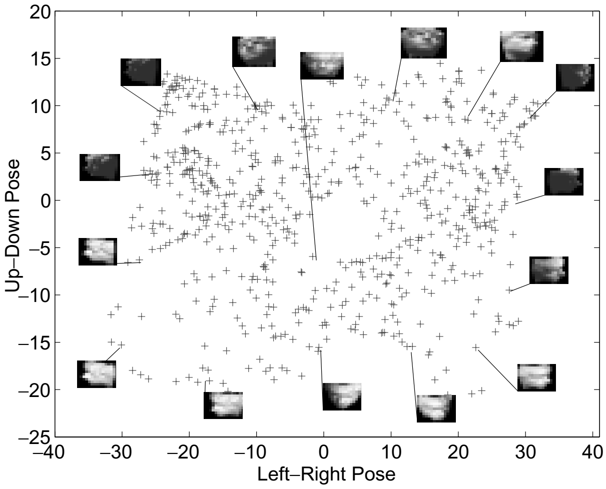

Fig. 6. Isomap computed two dimensional embedding of face database with 698 images. The top two degrees of freedom of the dataset corresponding to the pose of the face are clearly captured in the low dimensional representation. Images on the left and right halves of the embedding correspond to a leftward and rightward poses of the face respectively. Similarly, images on the top and bottom halves of the embedding correspond to an upward and downward poses.

both cases. Sequential-Isomap is marginally more expensive due to the added costs in performing online SVD. The average cpu time consumed for each additional sample in Sequential-Isomap is about 0.02 seconds on a 1.6 GHz Pentium III processor.

Figure 5 (*right*) compares the execution time ratios of the batch algorithms (Isomap, Landmark Isomap) and Sequential-Isomap. Approximately 6000 additional samples can be sequentially embedded into the manifold in the time required to perform a single batch Isomap computation with 2500 samples. The equivalent cost involved in the computation of a single batch Landmark Isomap embedding with 50 landmarks is comparable to sequentially embedding 900 samples.

9. Visual perception in human face data

High dimensional data often has a perceptually meaningful low dimensional structure that is not readily apparent in the original form. An example that has been extensively studied in the dimensionality reduction literature is an unorganized collection of images of a face acquired at various poses (i.e. left-right and up-down) and illumination conditions (http://isomap.stanford.edu). Isomap has been shown to reliably recover these dominant dimensions corresponding to pose and illumination in the computed low dimensional embedding.

Figure 6 shows the two dimensional embedding of 698 face images recovered by Isomap where all the points corresponding to the right half of the embedding physically correspond to the face posing right, while those on the left half correspond to the face posing leftwards. Similarly, the lower half of the embedding organizes the face posing downwards, while the upper half consists of images with an upward pose.

Figure 7 shows two dimensional embeddings computed by Sequential-Isomap for various choices of an initial training set, number of nearest neighbors and landmarks. The dominant dimensions of the data are clearly preserved in both the embeddings, though the aspect ratio is better preserved when $N = 250$. The quality of the solutions obtained with a small number of landmarks is especially encouraging. The

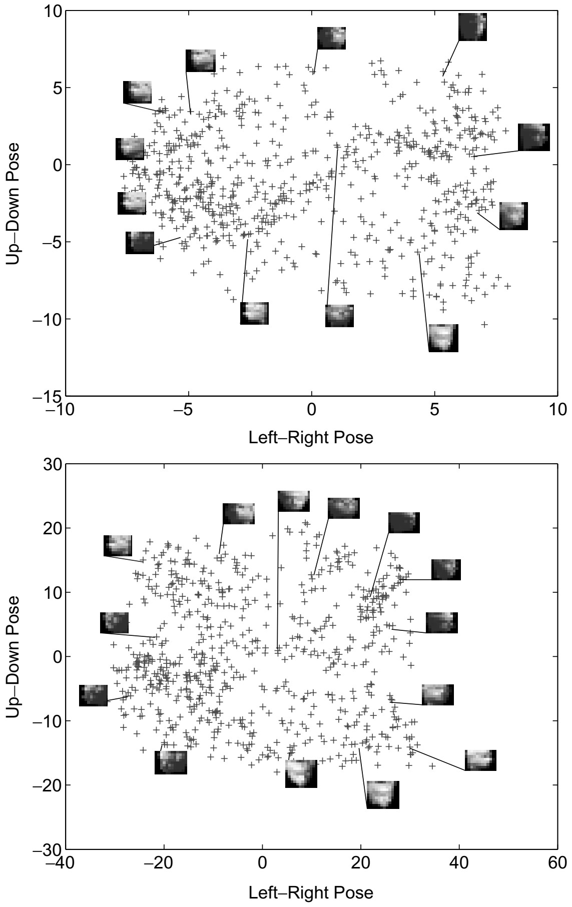

Fig. 7. Sequential-Isomap embedding of face database. Top ($N = 100, K = 16, L = 5$) and bottom ($N = 250, K = 5, L = 25$) where N, K, L are the combinations of the training set size, number of nearest neighbors and landmarks used in the computations. The embedding computed with $N = 250$ captures the underlying manifold geometry.

number of singular values that are updated online through incremental SVD equals the number of chosen landmarks; a small set of landmarks ensures lower computational costs.

10. Autonomous ground vehicle (AGV) dataset – Computation of low dimensional visual states

10.1. Introduction

Characterization of natural environments from sensory data plays a vital role in the navigation of autonomous systems. The utility of Sequential-Isomap in providing low dimensional descriptions of high dimensional image patches for subsequent use in an autonomous navigation task is investigated here.

10.2. Batch Landmark Isomap computations

A sample of 1690 high dimensional image patches representing colors and textures of typical groups in the environment such as sky, trees, bush and grass was selected from a large sequence of images

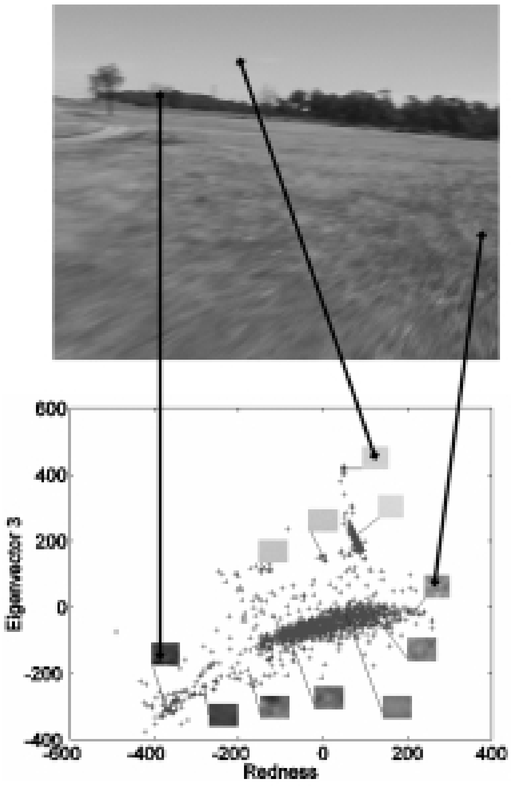

Fig. 8. Sample image acquired by the AGV (top) and low dimensional embedding of randomly sampled high dimensional image patches. Typical colours and textures in the environment are captured in the low dimensional representation. It is easily observed that darker shaded patches corresponding to the bush are grouped away from the lighter shaded patches corresponding to the sky.

acquired by an AGV at the Marulan test facility operated by the Australian Centre for Field Robotics. Texture information was included in the high dimensional input space by convolving each image with a bank of Gabor wavelets [8] at 2 scales and 2 orientations, resulting in an input space dimensionality of each 11 × 11 image patch of 847. Batch Landmark-Isomap with 45 landmarks was used to compute a low dimensional embedding of the training data.

Two of the top four dimensions of the computed embedding are shown in Fig. 8. It is readily observed that image patches corresponding to blue skies, bush and grassy patches are organized into reasonably distinct clusters. The batch embedding is used as the training set for Sequential-Isomap.

10.3. Sequential-Isomap embedding

In order to qualitatively evaluate Sequential-Isomap, the sample image (Fig. 8), top) was divided into 3380 image patches. Each of the image patches was used as sequential input to Sequential-Isomap. Two of the top four low dimensional coordinates corresponding to each image patch are displayed in the bottom row of Fig. 9. A batch Landmark-Isomap embedding was computed including all the points in the training set and the sample image, and the scaled low dimensional coordinates corresponding only to points in the sample image are shown in the top row of the same figure.

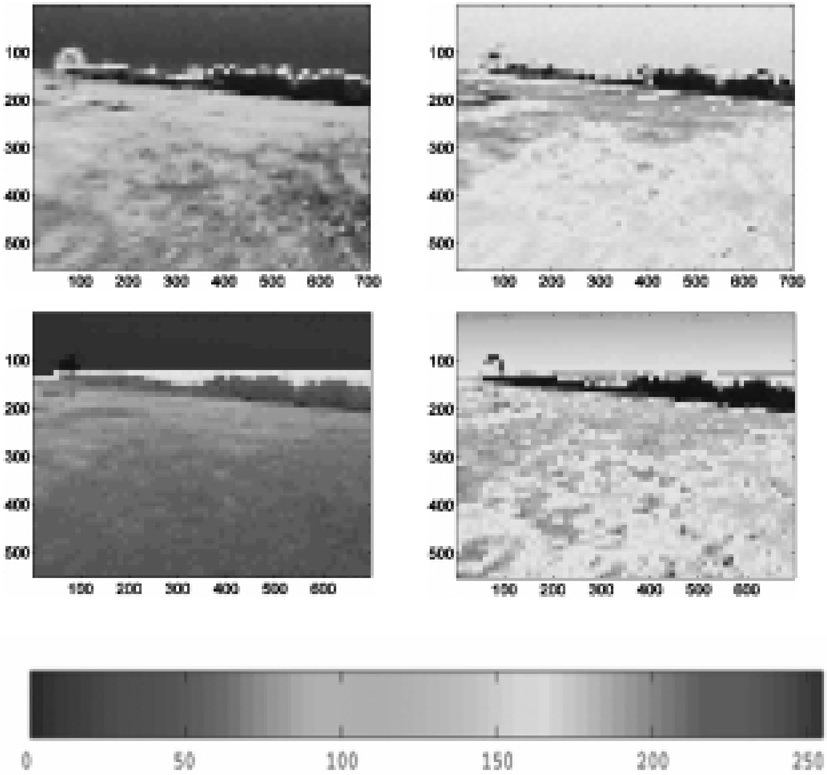

Fig. 9. Batch Isomap computed low dimensional states (top) displayed as grayscale contours on image patches. Objects with similar visual states are coded identically. The corresponding states are also computed by Sequential-Isomap (bottom). It is seen that there is good qualitative agreement between the batch and sequentially computed embeddings.

It is apparent from an examination of Fig. 9 that Sequential-Isomap with 1690 training samples accurately adapts the pre-computed manifold to the new samples obtained from the test image. The estimates of the top two states from the batch and incremental algorithms are very similar.

10.4. Practical implementation

An efficient implementation on an autonomous vehicle can be achieved by performing the PCA update asynchronously with respect to the incremental SVD. The frequency of the PCA computation must be balanced to achieve the conflicting objectives of near real time performance and maintenance of current sample states. In non-critical exploratory missions, the PCA update could be performed at infrequent intervals while the robot is stationary. Considering the small computational time of this step relative to a full batch Landmark Isomap solution (about 20 cpu minutes), efficient practical implementations can readily be devised.

11. Unmanned underwater vehicle (UUV) dataset – Computation of underlying visual states

In order to evaluate the generality of Sequential-Isomap in applications involving autonomous systems, data from a second environment was considered. A sample of about 8573 high dimensional points

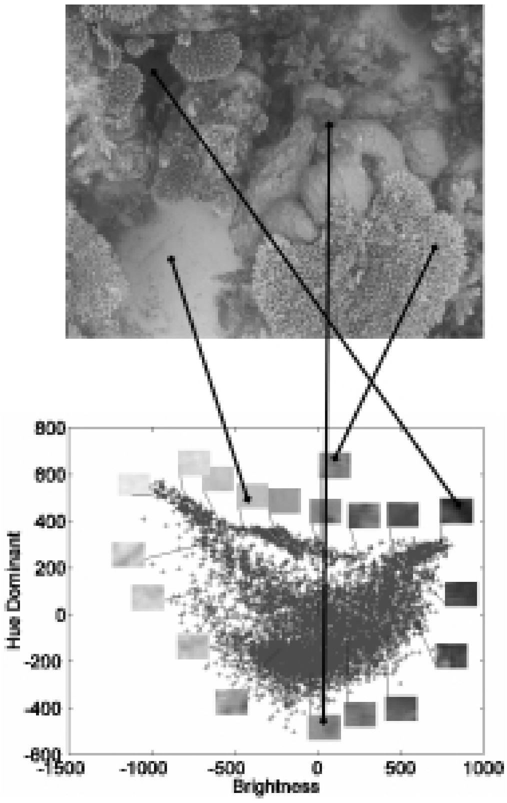

Fig. 10. Sample image acquired by the UUV (top) and low dimensional embedding of randomly sampled high dimensional image patches (bottom). Typical colours and textures in the underwater environment are captured in the low dimensional representation. The eigenvectors have been interpreted to correspond to the brightness and hue of the image patches.

physically representing colors and textures of typical objects in an underwater environment such as beach sand and corals was selected from a sequence of images [21] acquired from a camera mounted onto the UUV Oberon, operated by the Australian Center for Field Robotics.

Texture information that is vital in the characterization of the corals, was included in the high dimensional input space by convolving pixel patches with Gabor wavelets at 2 scales and 2 orientations [8] and batch Landmark-Isomap was used to compute a low dimensional embedding of the training data. Low dimensional embeddings are now computed through Sequential-Isomap and Landmark Isomap induction based on the sample image (Fig. 10, top) exactly as described in Section 8 and 9.

11.1. Sequential-Isomap and Landmark-Isomap induction

The results from Sequential-Isomap and Landmark-Isomap based induction are shown in Fig. 11. Unlike the texture impoverished environment explored by the AGV, the underwater data is rich in textures. It is unsurprising to observe that the third state (bottom row of Fig. 11) is correlated to high frequency textures of the coral colonies in the image.

The results from both approaches are strikingly similar due to the dense sampling of the manifold. The key difference between them is that Sequential-Isomap computes a low dimensional basis that includes all the samples that may be further expanded to incorporate new data onto the manifold. Landmark-Isomap

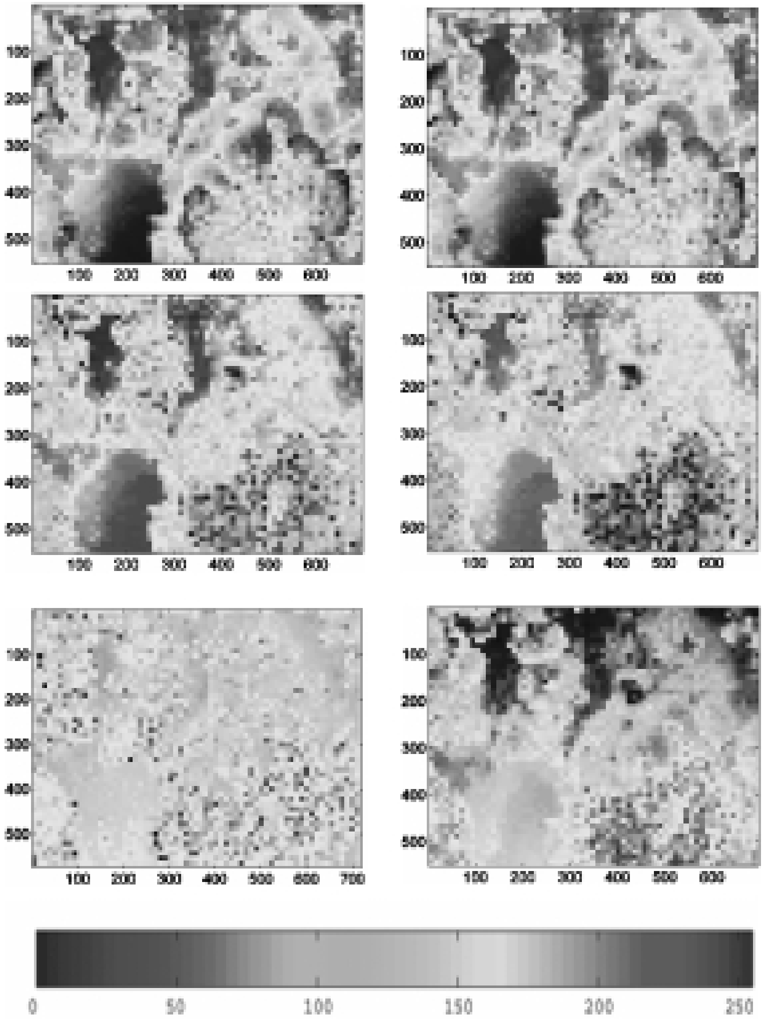

Fig. 11. Landmark Induction (left column) and Sequential-Isomap (right column) computed low dimensional states displayed as contours on the image plane. Top row – Low dimensional state corresponds to the brightness of the image patches. Middle row – Best correlated to the hue of the image patches. Bottom row – High frequency texture of the image patches.

based induction provides useful out-of-sample extensions for new data using a *fixed* basis comprised only of the training samples.

12. Conclusion

A sequential, unsupervised learning scheme Sequential-Isomap has been presented, that fuses together developments in incremental singular value decomposition and nonlinear dimensionality reduction. In principle, the concepts presented here could be used in conjunction with a range of NLDR algorithms to develop a whole new family of sequential, unsupervised learning algorithms. The particular combination of a modified Landmark Isomap scheme and incremental SVD presented here is extremely effective due to the resulting sparsity and thin rank of the SVD.

The accuracy of the algorithm has been numerically assessed through synthetic and real datasets, and in all cases, the underlying manifold geometry computed by a batch algorithm is faithfully replicated online. The convergence characteristics of the method have been numerically studied, and it is observed that the intrinsic embedding computed by batch Isomap is replicated as the number of training samples is progressively increased. In the context of real-time systems, a strong case has been made for further research directed towards applications of Sequential-Isomap.

Acknowledgments

This work is supported by the ARC Center of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales (NSW) State Government. The authors are grateful to the anonymous reviewers for their constructive comments.

References

[1] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Tech. Rep., University of Chicago, Department of Computer Science, 2002.

[2] Y. Bengio, P. Vincent, J. Paiement, O. Delalleau, M. Ouimet and N. Le Roux, *Spectral Clustering and Kernel PCA are Learning Eigenfunctions*, Tech. Rep., Department of Computer Science and Operations Research, University of Montreal, 2003.

[3] Y. Bengio J. Paiement and P. Vincent, Out of sample extensions to LLE, MDS and Isomap, in: *Advances in Neural Information Processing Systems*, (Vol. 16), 2003, pp. 682–688.

[4] M. Brand, Incremental singular value decomposition of uncertain data with missing values, Tech. Rep., Mitsubishi Electric Research Laboratory, 2002.

[5] J. Carrier, L. Greengard and V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, *SIAM Journal of Scientific and Statistical Computing* **9** (1988), 669–686.

[6] E.W. Dijkstra, A note on two problems in connexion to graphs, *Numerische Mathematik* **1** (1959), 269–271.

[7] V. DeSilva and J. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, in: *Advances in Neural Information Processing Systems, Becker* S. Thrun and K. Obermayer, eds, (vol. 15), 2002, pp. 705–712.

[8] D.J. Field, Relations between the statistics of natural images and the response properties of cortical cells, *Journal of the Optical Society of America* **4** (1987), 2379–2394.

[9] I. Foster, *Designing and Building Parallel Programs*, Addison Wesley, 1995.

[10] M. Gelb, *Applied Optimal Estimation*, MIT Press, 1974.

[11] G. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.

[12] V.K. Goyal and M. Vetterli, Block transform adaptation by stochastic gradient descent, in: *IEEE Digital Signal Processing Workshop*, 1998.

[13] M. Gu and S. Eisenstat, A stable and fast algorithm for updating the singular value decomposition, Tech. Rep., Department of Computer Science, Yale University, 1993.

[14] E.R. Jessup and D.C. Sorensen, A parallel algorithm for computing the singular value decomposition of a matrix, Tech. Rep., Argonne National Laboraroty – ANL-MCS-TM-102, 1991.

[15] S. Kumar, F. Ramos, B. Upcroft and H.F. Durrant-Whyte, A statistical framework for natural feature representation, in: *Proc. IEEE/RSJ Conference on Intelligent Robots and Systems, Edmonton, Canada*, 2005.

[16] T. Morita and T. Kanade, A sequential factorization method for recovering shape and motion from image streams, Tech. Rep., Carnegie Mellon University, School of Computer Science, 1994.

[17] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by Locally Linear Embedding," *Science* **290** (2000), 2323–2326.

[18] B. Scholkopf, A.J. Smola and K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10** (1998), 1299–1319.

[19] J. Tenenbaum, V. DeSilva and J.C. Langford, A global geometric framework for nonlinear dimensionality redution, *Science* **290** (2000), 2319–2323.

[20] J. Tenenbaum V. DeSilva and J.C. Langford, http://isomap.stanford.edu, 2000.

[21] S. Williams and I. Mahon, Simultaneous localization and mapping on the Great Barrier Reef, in: *Proceedings of the IEEE Conference on Robotics and Automation*, 2004.