Computer Vision and Image Understanding 116 (2012) 473–483

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Commute time guided transformation for feature extraction

Yue Deng, Qionghai Dai *, Ruiping Wang, Zengke Zhang

Automation Department, Tsinghua Nationality Laboratory for Information Science and Technology (TNList), Tsinghua University, China

ARTICLE INFO

Article history:
Received 15 January 2010
Accepted 8 November 2011
Available online 23 November 2011

Keywords:
Commute time
Random walk
Manifold learning
Face recognition
Feature extraction

ABSTRACT

This paper presents a random-walk-based feature extraction method called *commute time guided transformation* (CTG) in the graph embedding framework. The paper contributes to the corresponding field in two aspects. First, it introduces the usage of a robust probability metric, i.e., the commute time (CT), to extract visual features for face recognition via a manifold way. Second, the paper designs the CTG optimization to find linear orthogonal projections that would *implicitly preserve* the commute time of high dimensional data in a low dimensional subspace. Compared with previous CT embedding algorithms, the proposed CTG is a graph-independent method. Existing CT embedding methods are graph-dependent that could only embed the data on the training graph in the subspace. Differently, CTG paradigm can be used to project the out-of-sample data into the same embedding space as the training graph. Moreover, CTG projections are robust to the graph topology that it can always achieve good recognition performance in spite of different initial graph structures. Owing to these positive properties, when applied to face recognition, the proposed CTG method outperforms other state-of-the-art algorithms on benchmark datasets. Specifically, it is much efficient and effective to recognize faces with noise.

1. Introduction

Over last decades, many statistical and optimization models have been extensively used as a feature extraction step for pattern recognition and visual perception. Among these approaches, one prevalent paradigm is the subspace analysis, the aim of which is to preserve the statistical property of the high dimensional data in a low dimensional subspace [1]. Such methods include Principal Component Analysis (PCA) [2], Linear Discriminant Analysis (LDA) [3] and Non-negative Matrix Factorization (NMF) [4]. Although these typical algorithms have been successfully applied to many disciplines, they are based on linear statistical models that cannot be extended to reveal the non-linear relationships among data. Therefore, manifold learning methods are proposed to represent data distribution via a graph topology [5].

A manifold is locally linear, but globally non-linear. Different manifold learning algorithms have been proposed around the years; these include Isomap [6], locally linear embedding (LLE) [7], and Laplacian eigenmap (LE) [8]. However, most existing manifold algorithms cannot handle the out-of-sample data [9]. They only project the training samples on the graph into an embedding subspace but cannot generalize the learning results to test samples. Accordingly, these manifold embedding algorithms are effective clustering tools but are not appropriate algorithms for recognition.

In order to enhance the generalization ability of typical manifold learning algorithms, locality preserving projection (LPP) [9] was proposed as an extension of Laplacian eigenmap (LE) [8]. LPP is designed to preserve the locality of graph connections via a projection matrix. However, it could only preserve the locality of data while ignoring the relationship between samples that are not directly connected. Besides, although LPP is implemented on the graph, it also relies on the Euclidean distance between nodes.[1] Some recent works indicated that non-Euclidean metric on the manifold could reveal the essence of data much better [10].

As stated above, two challenges of manifold learning are: (1) how to generalize the learned mapping function to out-of-sample data that are not on the training graph and (2) how to find a robust metric/similarity to reveal the relationship of both the connected and non-connected nodes on a graph. To address these two points, in this paper, we propose a commute time guided (CTG) transformation for manifold learning.

In our CTG model, the commute time (CT) of random walk is adopted as a robust metric to evaluate the similarities between all pairs of nodes on the graph. CT is a probabilistic distance which records the average time required for a random walk to travel around a pair of nodes on the graph and thus is very robust to noise. Therefore, representing nodes affinities by CT relies less on

* Corresponding author. Address: Room 725, Central Main Building, Tsinghua University, Beijing 100084, China. Fax: +86 10 62788613.
E-mail address: qionghaidai@tsinghua.edu.cn (Q. Dai).

[1] In LPP, the local relationship between nodes are measured by some kernel functions, like Gaussian, embedded with Euclidean distance.

1077-3142/$ - see front matter Crown Copyright © 2011 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2011.11.002

the initial graph structure and is less sensitive to the noisy disturbances in the training data. Then, inspired by the framework of LPP [9] and graph embedding (GE) [1], an orthogonal linear projection matrix is optimized to *implicitly preserve* the commute time of the high dimensional manifold in a low dimensional subspace. The implicit preservation means that we are not going to strictly keep the same CT quantity in the mapping space. Instead, in the optimization, CT is used to guide the embedding that a large CT between nodes in the high dimensional space will induce a large Euclidean distance in the mapped space and vice versa.

Owing to the generalization ability of CTG model, we can extend the power of CT to recognition tasks. In this paper, we apply the CTG model to recognize faces. The performance of the proposed method will be evaluated on different graph topologies and compared with other graph similarities. Besides, on four benchmark face datasets, the CTG will be compared with other state-of-the-art recognition algorithms. It is concluded that the CTGface extracted via commute time is an efficient and effective feature for face recognition. Moreover, the CTG method is especially robust and efficient to recognize faces with noise.

1.1. Related work

Although commute time for data embedding is a long standing topic in computer vision, our approach is quite different. Previous works [11,12] first construct the data graph and compute an affinity matrix recording the commute time between each pair of nodes. Then, the commute time matrix is decomposed in terms of spectral embedding techniques, e.g., ratio cut [13] or normalized cut [14].

Spectral embedding guarantees that original commute time will be preserved in the embedding space. In [12,11], CT embedding technique has been proven to be a specific form of kernel PCA by CT kernels. In [11], Qiu and Hancock revealed the relationship of CT embedding and LE. The classic CT embedding algorithm has been successfully applied to many practical tasks including image segmentation [15], motion segmentation [11] and image representation [16]. But in all these interesting applications, the CT embedding technique is only adopted as a powerful tool for data clustering.

Different from previous CT embedding algorithms, our CTG has the following advantages. First, it is a graph-independent algorithm. CTG has the potential to generalize the learning ability to the out-of-sample data. Besides, it is less sensitive to the initial graph structure. Experimental results (see Section 4.2) verify that CTG outperforms other benchmark algorithms on different graph topologies, e.g. KNN graph and sparse graph. Finally, it is possible to explicitly define a closed-form solution to the CTG optimization in terms of eigen-decomposition, which greatly speeds up the training procedures.

An early conference version of this paper has been published in [17]. It just provided the preliminary idea of the random walk for face recognition. In this paper, we enhance the discussions from both the theoretical and experimental perspectives. First, a thorough and rigorous mathematical formulation on CTG optimization is provided in the paper. The formulation in [17] is only based on a one dimensional projection vector. Then, the meaning of the 1D projection is extended to a general matrix case. In this paper, the formulation on CTG optimization is unified into a matrix framework of trace computation, which is more general and rigorous. Besides, the experimental discussions are greatly enhanced. In [17], the experiments were only based on a KNN graph. In this paper, the experiments are conducted on more graph topologies and the commute time metric is compared with other graph metrics. Moreover, we compare our method against a number of state-of-the-art competitors on more benchmark datasets.

1.2. Organization

The remainder of this paper is organized as follows. We first introduce the formulation of commute time and discuss its interesting properties in Section 2. Then, the commute time guided transformation will be proposed in Section 3. Discussions and experiments of random walk for face recognition are conducted in Section 4. Section 5 concludes the paper and provides some discussions about future works.

2. Commute time and its properties

Before introducing the proposed CTG model, in this section, we will first give the definition of commute time (CT) and provide some discussions about its interesting properties that are mostly related to manifold learning.

2.1. Formulation of commute time

The calculation of commute time is an old topic in the areas of applied mathematics, physics and the field theory. Without loss of generality, in this paper, we adopt a ***Markov*** based calculation as it is the most straightforward one.

We start the introduction from a probabilistic random walk, which is mainly based on the graph topology. Accordingly, we define a weighted, undirected graph G with a symmetric weight W_{ij} for the edge between nodes i and j. The value of W_{ij} represents the degree of affinity between the two nodes. $W_{ij} = 0$ means that there is no direct connection between them. There are many ways to construct such a graph, we will elaborately discuss them in the experimental part.

The probability that a random walk travels from node i to its connecting node j is defined via

$$p_{ij} = \frac{W_{ij}}{\sum_t W_{it}}, t \in \theta(i), \tag{1}$$

where $\theta(i)$ is a set containing all the nodes connected to node i. From the definition of travelling probability p_{ij}, it is obvious that the larger the connecting weight is, the more probably the random walk will travel via this way.

Due to the probability behavior of random walk, one critical problem comes out inevitably. A random walk could follow different paths to travel between a pair of nodes, and the corresponding time cost could be quite different. Fortunately, the commute time is a statistical expectation, which is a fixed value. It represents the average time that a random walk travels around a pair of nodes. The commute time is related to the global structure of the graph rather than a single path or local connections only [18].

In this paper, we mainly follow the results in [19] which indicated that the formulation of the commute time between a pair of nodes i and j, i.e., ct_{ij} is defined as:

$$ct_{ij} = volG \times \left(l^+_{ii} + l^+_{jj} - 2l^+_{ij}\right), \tag{2}$$

where $volG = \sum_{ij} W_{ij}$ is the *volume* of a graph, and $l_{ij} = [\mathbf{L}^+]_{ij}$ in which $\mathbf{L}$ is the Laplacian matrix and $(\cdot)^+$ stands for Moore–Penrose general inverse. The definition of $\mathbf{L}$ for a graph is given as follows:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{3}$$

where $\mathbf{W}$ is the *weight matrix* and $\mathbf{D}$ is the *degree matrix* in the spectral graph theory. The degree matrix is defined as:

$$\mathbf{D} = diag\left(\sum_j W_{ij}\right), \tag{4}$$

where *diag* is the diagonal operation.

2.2. Property of commute time

Based on the introduction of commute time above, some of its positive properties for manifold learning can be summarized.

1. Different from traditional ℓ_p norm based distance, commute time is calculated from a probabilistic model. It describes the relationship of data by considering all the feasible paths between them. Therefore, commute time could break the restriction of traditional Euclidean norm and can thus reveal the manifold distribution of data much better [11].
2. Commute time of random walk is a very robust metric. It is not related to any single path on the graph. Thus, it is robust to noise.
3. Compared with other graph distances, for example the geodesic distance, the calculation of commute time is much more efficient which just requires to solve a Moore–Penrose general inverse problem (see Eq. (2)).

These positive properties naturally facilitate CT be an ideal metric for manifold learning.

3. Commute time guided transformation

While CT has those positive properties as above, most existing algorithms on CT embedding are graph-dependent that can only handle the data on the initial training graph [11]; and therefore are only suitable for applications related to data clustering. But there are many other computer vision tasks beyond the scope of data clustering, e.g. recognition. Can we take the advantage of the robustness of the CT metric and, meanwhile, generalize the learning ability to the out-of-sample data? In this section, we will propose a graph-independent CT embedding algorithm called *commute time guided* (CTG) transformation.

3.1. Commute time preserving strategy

In most embedding optimizations, the prominent purpose is to map high dimensional data into a low dimensional subspace, in which the original metric is preserved. Before explaining our algorithm, we will first define some notations. We define the node on the graph as $\mathbf{N}_i \in \mathbb{R}^{M\times 1}$. Commute time between $\mathbf{N}_i$ and $\mathbf{N}_j$ is defined as ct_{ij}. $\mathbf{y}_i \in \mathbb{R}^{m\times 1}$ represents the data in the transformed subspace, $m \ll M$. The Euclidean distance between data $\mathbf{y}_i$ and $\mathbf{y}_j$ in the subspace is $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$.

As stated above, the general idea behind embedding techniques is based on the preservation strategy. How to preserve the commute time in an embedding space? To address this issue, the most straightforward approach is the quantity preservation strategy. The quantity preservation means that the numerical quantity of the Euclidean distance in the subspace should be the same as the original commute time. The optimization can be designed as:

$$(\mathrm{MDS}-\mathrm{CT})\ \min . \sum_{ij} \|d_{ij} - ct_{ij}\|^2. \tag{5}$$

(5) is an *explicit preservation* strategy since the optimal solution to it, if can be found, is $d_{ij} = ct_{ij}$. The explicit preservation method sounds reasonable, however, it is not suitable for the recognition task here. Essentially, the optimization in (5) is the classical Multi-dimensional Scaling (MDS) [20]. It is the same as the framework used in the Isomap [6]. The only difference is that Isomap preserves the geodesic distance instead of the CT. Although this MDS framework is straightforward, it has two significant drawbacks.

First, as indicated in [20,21] and many other related works, there is no closed-form solution to the MDS optimization. Many known algorithms to solve it are based on iterative approaches which is computationally expensive; and moreover is likely to get stuck in local optimum. The second drawback of MDS framework is the same as existing commute time embedding techniques, i.e., it is graph-dependent. The results of MDS are the coordinates of data (i.e. y_i and $\mathbf{y}_j$) in the embedding space. To the best of our knowledge, it cannot be generalized to the data that are not on the initial graph.

Due to these two reasons, we will not adopt the MDS framework to preserve the commute time in the subspace. Accordingly, we design a CTG optimization that *implicitly preserve* the CT in the subspace. The implicit preservation means that we are not going to strictly keep the same CT quantity in the mapped space. Instead, the rank or concordance[2] of CT are preserved in the mapped space.

3.2. Commute time guided transformation

According to previous discussions, we propose the CTG objective by using the commute time as a measure of affinity between two nodes:

$$\min \sum_{ij} \frac{d_{ij}^2}{ct_{ij}} = \min \sum_{ij} \frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{ct_{ij}} \tag{6}$$

The meaning of (6) is self-evident: ct_{ij} is a penalty term. If ct_{ij} is small, then, d_{ij} should also be small enough to minimize the whole objective. A small ct_{ij} with a large d_{ij} may contribute a large quantity to the objective function which greatly affects the global minimization. On the contrast, if ct_{ij} is large, it allows a comparable large d_{ij} in the mapped space. According to these discussion, it is clear that the value of commute time is used as a penalty to guide the optimization from graph to the subspace, which is the main reason that we call it commute time guided (CTG) transformation.

Moreover, in order to generalize the learning ability of CTG to unknown samples, a unitary projection matrix $\Omega \in \mathbb{R}^{M\times m}$ is introduced. Ω^T maps the node $\mathbf{N}_i \in \mathbb{R}^M$ to a point $\mathbf{y}_i \in \mathbb{R}^m$ in the subspace, i.e., $\mathbf{y}_i = \Omega^T\mathbf{N}_i$. Accordingly, we get the objective function[3] of CTG model,

$$(\text{CTG Obj.})\ \min_{\Omega\in\mathbb{R}^{M\times m}} \sum_{ij} \frac{\|\Omega^T\mathbf{N}_i - \Omega^T\mathbf{N}_j\|^2}{ct_{ij}} \tag{7}$$

We write the objective in (7) in the form of matrix computation by the trace term:

$$\begin{aligned}\sum_{ij} \frac{\|\Omega^T\mathbf{N}_i - \Omega^T\mathbf{N}_j\|^2}{ct_{ij}} &= \sum_{ij} \frac{1}{ct_{ij}} tr[(\Omega^T\mathbf{N}_i - \Omega^T\mathbf{N}_j)(\Omega^T\mathbf{N}_i \\ &\quad - \Omega^T\mathbf{N}_j)^T] \\ &= tr\left[\sum_{ij} \frac{(\Omega^T\mathbf{N}_i - \Omega^T\mathbf{N}_j)(\Omega^T\mathbf{N}_i - \Omega^T\mathbf{N}_j)^T}{ct_{ij}}\right] \\ &= 2tr\left[\sum_i \frac{\Omega^T\mathbf{N}_i\mathbf{N}_i^T\Omega}{ct_{i.}} - \sum_{ij} \frac{\Omega^T\mathbf{N}_i\mathbf{N}_j^T\Omega}{ct_{ij}}\right] \\ &= 2tr(\Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T\Omega)\end{aligned} \tag{8}$$

where $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2, \ldots, \mathbf{N}_n]$ (n is the number of nodes on the graph); $\mathbf{G} = [g_{ij}] = [1/ct_{ij}]$ ($\mathbf{G} \in \mathbb{R}^{n\times n}$), and ct_{ij} is the commute time between nodes i and j; $ct_{i.} = \sum_j ct_{ij}$ and therefore, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose entries are the sums of the columns (or the rows since $\mathbf{G}$ is symmetric) of $\mathbf{G}$. $tr(\cdot)$ is the trace of a matrix and we know that for

[2] We will explain the meaning of concordance in the next subsection.

[3] It is only the objective function of the CTG optimization. The whole CTG optimization comes with a constraint. The reason for the constraint will become clear latter.

any matrix $\mathbf{P}$, $\|\mathbf{P}\|^2 = tr(\mathbf{P}\mathbf{P}^T)$. The second equality in (8) holds because, for any scalar α_{ij} and any matrix $\mathbf{Q}$, $\sum_{ij}\alpha_{ij}tr(\mathbf{Q}) = \sum_{ij}tr(\alpha_{ij}\mathbf{Q}) = tr\left[\sum_{ij}(\alpha_{ij}\mathbf{Q})\right]$

Although we gave the matrix explanation to the objective function, it cannot be directly minimized. It is because that if no extra constraint was added, the objective in (6) will map all the data to the same point, i.e. $\mathbf{y}_i = \mathbf{0}$, $\forall i$. Therefore, some regularization terms are needed to avoid the trivial solution. We follow the general idea in LE [8], LPP [9] and GE [1] to add such a constraint.

In the transformation (8), matrix $\mathbf{A}$ provides a measure of the relative importance of training samples. The larger the element $\mathbf{A}_{ii}$ is, the more important the node $\mathbf{y}_i$ will be for the final solution [9]. Following the same constraint in [9], in our formulation, we also impose a constraint as follows:

$$\Omega^{\mathrm{T}}\mathbf{N}\mathbf{A}\mathbf{N}^{\mathrm{T}}\Omega = \mathbf{I}, \tag{9}$$

where $\mathbf{I}$ is the identity matrix. Therefore, the CTG optimization is described as follows:

$$\begin{aligned}(\mathrm{CTG})\ &\min_{\Omega\in\mathbb{R}^{m\times M}} tr[\Omega^{\mathrm{T}}\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^{\mathrm{T}}\Omega]\\ &\text{s.t. } \Omega^{\mathrm{T}}\mathbf{N}\mathbf{A}\mathbf{N}^{\mathrm{T}}\Omega = \mathbf{I},\end{aligned} \tag{10}$$

This is a problem of constraint optimization, which can be solved using the Lagrange Multiplier [22],

$$L(\Omega,\Lambda) = tr(\Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T\Omega) - \langle\Lambda,(\Omega^T\mathbf{N}\mathbf{A}\mathbf{N}^T\Omega-\mathbf{I})\rangle \tag{11}$$

where Λ is the lagrangian multiplier, which is a diagonal matrix $\Lambda\in\mathbb{R}^{m\times m}$. m is the dimension of data in the embedding space. $\langle\cdot\rangle$ is an inner product of two matrices. The optimal solution to the projection can be obtained by setting $\nabla_\Omega L(\Omega,\Lambda)=\mathbf{0}$. After some derivations (see Section A), the minimization of (11) reduces to a generalized eigen-decomposition:

$$\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^{\mathrm{T}}\Omega = \Lambda\mathbf{N}\mathbf{A}\mathbf{N}^{\mathrm{T}}\Omega. \tag{12}$$

According to the optimization theory, with the objective of minimizing the above equation, those eigenvectors corresponding to the smallest eigenvalues are selected as the linear projections. Besides, the objective in (10) is convex and thus the minimum is the global optimum. These projection vectors are utilized to extract features for recognition.

3.3. Preservation verification

As stated in the last subsection, the CTG optimization could implicitly preserve the commute time in the subspace. In this part, we will perform some empirical observation to verify that the proposed model is effective enough to represent the commute time using the squared Euclidean distance. For each ct_{ij} on the original graph, we can get its corresponding squared Euclidean distance d^s_{ij} in the embedding space. We denote such a correspondence as (ct_{ij}, d^s_{ij}). Suppose there are n nodes on the graph, we can get $\frac{1}{2}n(n-1)$ correspondences. In order to verify that the CTG model can well preserve the commute time in the subspace, we use the AR face dataset [23] to conduct experiments.

AR dataset: The AR dataset consists of over 4000 frontal images for 126 individuals. For each individual, 26 pictures were taken in two separate sessions. These images have facial variations caused by illumination change, expressions, and occlusion. All the face images are normalized to a resolution of 64×64 pixels based on the eye locations, and color images are converted to grayscale ones. In order to enhance the global contrast of the images and reduce the effect of uneven illumination, histogram equalization is applied to all the images.

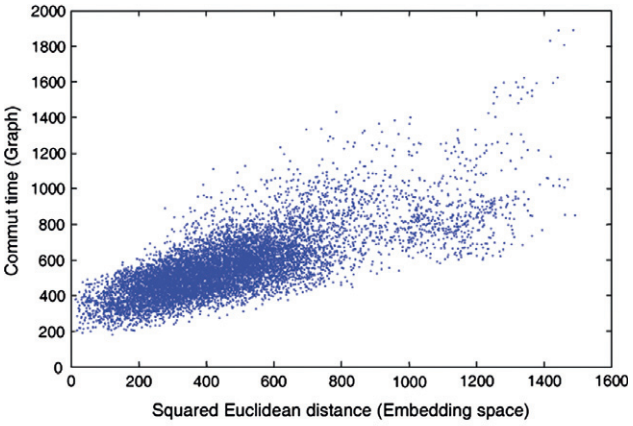
In the experiment, we randomly select two faces of each subject in AR dataset as nodes on the training graph. The graph is spanned using KNN method. Please refer to Section 4.2 for detailed discussions about the initial graph topology. Then, CTG optimization is performed on the graph and all the nodes are projected into the subspace. Fig. 1a plots all the correspondences of the commute time and the squared distance. The dimensions of the embedding space is $m = 20$. From the figure, it is observed that the commute time and the squared distance have high correlations. Besides, (ct, d^s) correspondences densely distribute on the left-down part in Fig. 1a. It is because that most CTs on the initial training graph are in this area. In order to further investigate such correlations, some quantitative analysis will be performed.

First, we calculate the widely used linear correlation coefficient $\gamma_{ct,\ dis}$ of the two sequences:

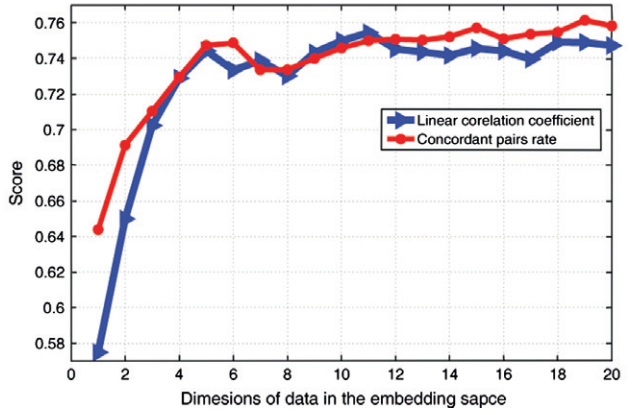
$$\gamma_{ct,dis} = \frac{\sum_{ij}(ct_{ij}-\overline{ct})\left(d^s_{ij}-\overline{d^s}\right)}{\sqrt{\sum_{ij}(ct_{ij}-\overline{ct})^2\sum_{ij}\left(d^s_{ij}-\overline{d^s}\right)^2}}$$

Besides, as stated above, the CTG model only implicitly preserves the original commute time. It means that a large commute time (ct_{ij}) on the graph will induce a large squared distance d^s_{ij} in the subspace and vice versa. To judge this critical property, we define the concordant pairs rate.

Definition 3.1. Let $S = \left\{\left(ct_{ij}, d^s_{ij}\right), 1\leqslant i\leqslant n, \frac{n}{2}+1\leqslant j\leqslant n\right\}$ be a set recording all the commute time and squared distance correspondences. Any pair (ct_{mn}, d^s_{mn}) and $\left(ct_{pq}, d^s_{pq}\right)$ are said to be concor-



(a) Commute time and squared Euclidean distance pairs.



(b) Quantitative evaluations

Fig. 1. Experimental verifications for preservation effectiveness of CTG optimization.

dant if the ranks for both elements agree, that is: if $ct_{mn} > ct_{pq}$ then $d^s_{mn} > d^s_{pq}$; or if $ct_{mn} < ct_{pq}$ then $d^s_{mn} < d^s_{pq}$. They are said to be discordant, if $ct_{mn} > ct_{pq}$ and $d^s_{mn} < d^s_{pq}$ or if $ct_{mn} < ct_{pq}$ and $d^s_{mn} > d^s_{pq}$.

We denote C as the number of concordant pairs in the set S and D as the number of discordant pairs.The concordant pairs rate ρcan be computed via

$$\rho = \frac{C}{C+D}$$

These two quantitative scores are reported in Fig. 1b, where the dimension of data in the subspace, i.e. m, varies from 1 to 20. From the result, it is obvious that both the linear correlation coefficient and the concordant pairs rate are more than 0.74 (maximal 1), which demonstrate that the squared distance the original commute time have high correlations.

From both the empirically observation (Fig. 1a) and the quantitative analysis (Fig. 1b), it is concluded that the proposed CTG model is an effective projection strategy to implicitly preserve the original commute time in the subspace. In the next section, thorough experiments on face recognition will be conducted to verify the feature extraction functionality of CTG method to out-of-sample data.

4. Random walk for face recognition

In the previous part, we have proposed the CTG method for dimensionality reduction. In order to verify the effectiveness of the proposed method, in this part, the CTG will be used to extract features for face recognition.

4.1. Face recognition using the CTG feature

In order to use CTG for face recognition, a graph topology should be first constructed. On the graph, the nodes are from faces in the training set, and the connections between nodes are established. There are a number of methods for graph construction. However, in this part, we will omit them. The detailed discussions on graph constructions will be extended in the next subsection.

After the construction of the graph, the CTG method is implemented on the face graph to extract the projection matrix Ω. Fig. 2 illustrates the optimal projections derived by using the eigenfaces [2], the fisherfaces [3], the Laplacianfaces [9], and the CTGfaces, respectively, based on the Yale face dataset. Our proposed CTGfaces have similar appearances to the Laplacianfaces, because both our algorithm and LPP are based on manifold.

For recognition purpose, both the training and testing faces are projected into the subspace via Ω. The low dimensional points in the subspace are referred as features for face recognition. These features are identified by a nearest-neighbor classifier. The complete procedures of our face recognition algorithm are listed in Algorithm 1. In the algorithm, *Nearest*($\mathbf{y_i}$, $\mathbf{F}$) is a function which returns the label ℓ_i for the i^{th} testing face using the nearest-neighbor classifier.

Algorithm 1. Face recognition using CTG features

Input: Training faces set $\mathbf{N} = \{\mathbf{N}_1, \mathbf{N}_2, \ldots, \mathbf{N}_n\}$ and testing faces set $\mathbf{f} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_t\}$

Training:

1. Span $\mathbf{N}$ as a graph $\Gamma = \{\mathbf{N}, \mathbf{W}\}$;
2. Compute the commute time matrix **CT** using (2);
3. Compute matrix **G** with each entry $G_{ij} = 1/ct_{ij}$ and matrix $\mathbf{A} = diag\left(\sum_j G_{ij}\right)$;
4. Solve eigen-decomposition problem in (12) and get the optimal projection matrix Ω;
5. Embed nodes set **N** into the subspace and get the feature set $\mathbf{F} = \{\mathbf{F}_i = \Omega^T \mathbf{N}_i,\ i = 1,2,\ldots,n\}$;

Recognition:

6. **for** $k \leftarrow 1$ **to** t **do**// t `is the number of faces to be recognized`
7. // $\boldsymbol{y}_k$ `is the feature.`
8. Embed test faces into the subspace $\mathbf{y}_k = \Omega^T \mathbf{f}_k$;
9. $\ell_k \leftarrow$ `Nearest` y_k, F// label the testing face.
10. **end**

Output: Recognized face labels ℓ

4.2. Graph topology and graph similarity discussions

As discussed above, the training procedures of the CTG method rely on a graph topology. Therefore, before performing the proposed CTG method on benchmark datasets for face recognition, in this part, we will first discuss which graph topology is most suitable for CTG learning. There are a number of graph constructing methods and we will review some widely used ones, i.e., K-Nearest-Neighboring (KNN), Gaussian KNN (G-KNN) and ℓ_1 graph [24]. ℓ_1 graph is also known as sparse graph.

KNN graph and G-KNN graph are two typical methods for graph construction. KNN graph considers that one node only connects with its nearest k neighbors with the connecting weights defined as one. G-KNN is an extension of the KNN method, which uses a Gaussian kernel to penalize the Euclidean distance between two nodes. With the recent progresses of compressed sensing [25], a novel concept on constructing graph with sparse representation has been proposed. It considers that each node on the graph can be represented by all the other nodes via sparse classification. The basic formulation to construct the sparse graph is given as:

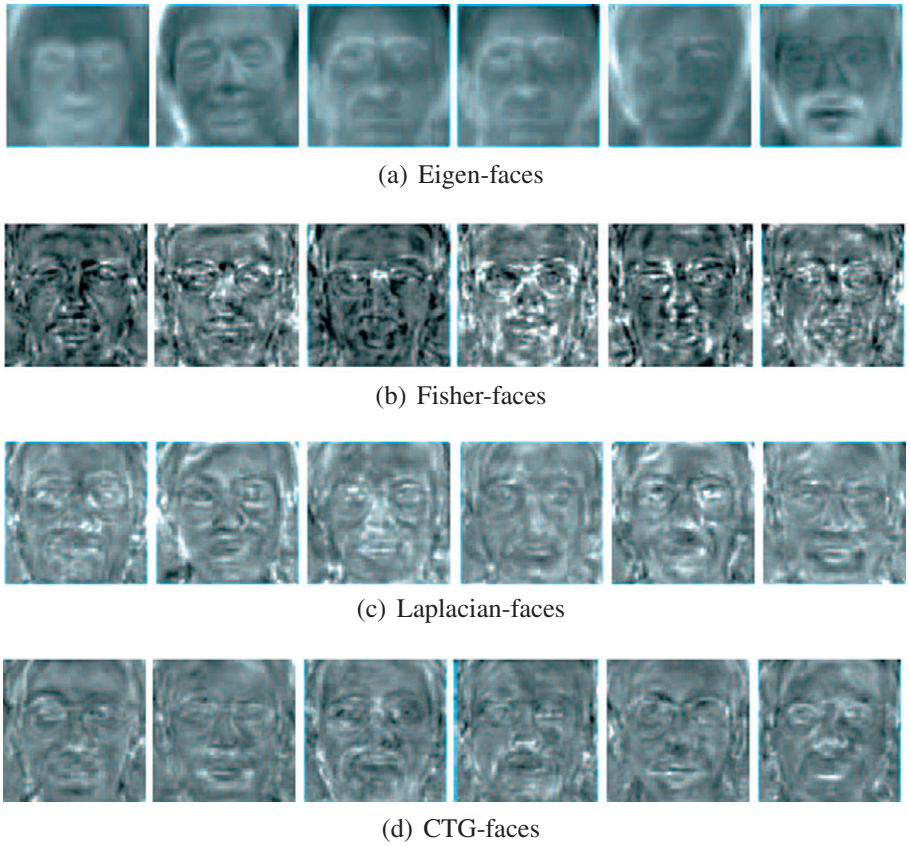
$$\begin{aligned} \min \quad & \|\alpha^{(i)}\|_{\ell_1} \\ \text{s.t.} \quad & \mathbf{N}_i^T = \alpha^{(i)} \mathbf{N}_{r(i)}, \end{aligned} \tag{13}$$

where $\|\cdot\|_{\ell_1}$ is the ℓ_1 norm [25]; $\mathbf{N}_i \in \mathbb{R}^M$ represents the i^{th} node, $\mathbf{N}_{r(i)} \in \mathbb{R}^{(n-1)\times M}$ stands for a matrix that is stacked by all the nodes on the graph excluding node i; and $\alpha^{(i)} \in \mathbb{R}^{1\times(n-1)}$ is a sparse approximation , in which the positive solutions will lead to weight connections, i.e., if $\alpha^{(i)}_j > 0$, then, $W_{ij} = 1$. Sparsity Induced Graph (SIG) is an extension of the unweighted sparse graph, which denotes the weights between nodes via a Sparseness Induced Similarity (SIS) [26]. The weights of SIG is defined via

$$W_{ij} = \frac{\max\left\{\alpha^{(i)}_j, 0\right\}}{\sum_{k=1}^{n-1} \max\left\{\alpha^{(i)}_k, 0\right\}}.$$

We have stated four graph topologies: two weighted graphs (G-KNN and SIG) and two unweighted graphs (KNN and SG). In order to justify which graph topology is the best one for face recognition, we will perform the CTG methods on these four graph topologies, respectively. Besides, the commute time will be compared with other graph similarities, e.g. locality similarity (LPP) and geodesic similarity (Isomap). For locality similarity, the LPP method is directly perform on different graph topologies. The geodesic projection (GEO) can be derived from (7), where the commute time between nodes (ct_{ij}) is replaced by the geodesic distance between them. The experiments are based on two public face datasets, i.e. the AR dataset [23] (has been introduced previously) and the FERET dataset [27].

The FERET face recognition dataset is a set of face images collected by NIST from 1993 to 1997. There are more than 1100 subjects in the FERET dataset. In each subject, the faces are captured



(a) Eigen-faces

(b) Fisher-faces

(c) Laplacian-faces

(d) CTG-faces

Fig. 2. The first six projections extracted from the Yale dataset based on (a) PCA, (b) LDA, (c) LPP, and (d) CTG.

via different poses, under various illuminations and with different expressions. In our experiment, we only use the frontal faces and the faces whose pose angles are less than 15°. Prior to processing, the faces are registered to each other based on the eye locations, and are normalized to the resolution of 64×64 pixels. The illumination histogram equalization is applied to all the images.

In the experiment, we randomly select half of faces in AR dataset and FERET dataset as training samples. The other half of faces in each dataset are treated as testing samples. Randomly choosing the training set ensures that the results and conclusions will not depend on any special choice of the training data. We follow steps in Algorithm 1 to conduct experiments on face recognition and it is repeated for 10 times. The average recognition rates on different graph topologies with different graph similarities are shown in Fig. 3.

In the AR dataset, commute time outperforms geodesic distance and locality similarity on KNN, GKNN, Sparse Graph and Sparseness induced graph. Commute time achieves the recognition rate of 80.2%, 76.3% and 81.2% on these three graph topologies, respectively. However, on the SIG, geodesic distance is the best one whose recognition rate is 73.2%, which only makes the improvements of 0.5% on commute time metric. The detailed comparisons on AR dataset are shown in the left part of Fig. 3. Among all the graph results, the highest recognition rate on AR dataset is achieved by the commute time on the sparse graph (81.2%). The second high recognition rate is obtained by random walk on the KNN graph (80.2%).

In the FERET dataset, the commute time based method is the best one on all the four kinds of graph structures. It achieves the recognition rate of 78.3%, 72.3%,79.4% and 77.7%, respectively. Geodesic distance gains similar performance as commute time, the recognition rates of which are 72.1%, 71.2%, 79.3%, and 76.2%, respectively. Both these two metrics outperform the locality similarity. The highest recognition rate is also achieved on the sparse graph with random walk. The recognition results on FERET dataset are shown in the right part of Fig. 3, based on which, some discussions on graph similarities and graph topologies are extended.

4.2.1. Graph similarity

Commute time and geodesic distance outperform the locality similarity on the face recognition test. It may be ascribed to that the locality similarity only represent the local relationship of connecting nodes. However, the commute time and geodesic distance could reveal both the local and global similarities of nodes no matter whether they are connected or not.

Commute time and geodesic distance achieve similar recognition performances. However, compared with geodesic distance, the commute time owns one prominent advantage. The calculation of commute time is much more efficient than the calculation of geodesic distance. The calculation of geodesic distance is mainly based on the greedy search, which is quite expensive. But the calculation of commute time just requires to solve a general inverse problem (see Eq. (2)). On the face manifold spanned by faces in AR dataset, it requires more than 47 s to calculate the geodesic distances between all pairs of nodes. and only 6 s to compute the commute time.

4.2.2. Graph topology

Among these four graph topologies, sparse graph is the best one for manifold based face recognition. Almost all the three graph similarities achieve their highest recognition rates on the sparse graph. However, there is one significant drawback of the SG. The construction of SG graph is too much time consuming, which

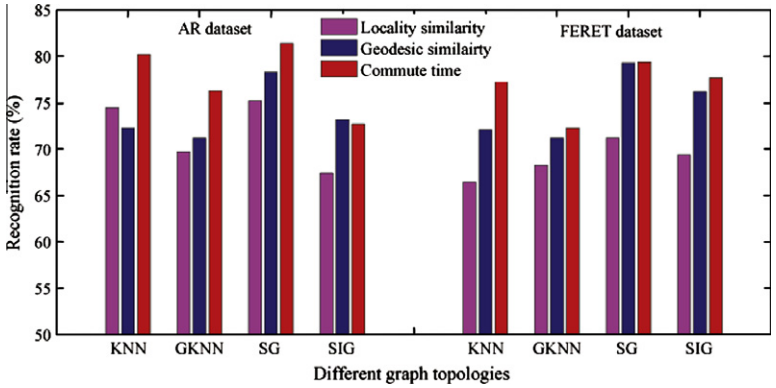


Fig. 3. The comparisons of commute time with different graph similarities on different graph topologies: K-Nearest-Neighbors (KNN), Gaussian KNN (GKNN), Sparse Graph (SG) and Sparseness Induced Graph (SIG).

requires to solve a convex ℓ_1 minimization for all the nodes. In the AR dataset, it requires as many as 1267 s (21 min) to construct the sparse graph. However, for the KNN graph, the graph topology can be constructed in 12 s.[4] Besides, the performances on the KNN graph is also comparable good, which is better than GKNN and SIG. Therefore, it makes a conclusion here that if one would like to get the best recognition performance, the sparse graph is recommended. However, by taking both the effectiveness and the efficiency into consideration, the KNN graph is the most suitable one for feature extraction.

4.3. General evaluations on benchmark datasets

In the preceding subsection, we have discussed the graph topologies and some graph similarities for face recognition. It is found that the random walk on the sparse graph or on the KNN graph achieves better recognition performances. In this part, we will extend these findings to practical face recognition tasks. The proposed CTG will be compared with other state-of-the-art methods for face recognition, e.g., PCA [2], LDA [3], NMF [4], Sparse Representation (SR) [28], LPP [9] and geodesic projection (GEO). Four standard face datasets used in the experiments are: the Yale dataset [29], the AR dataset [23], the PIE dataset [30], and the FERET dataset [27]. Here, we will only introduce the Yale and PIE datasets because the AR and FERET datasets have been introduced previously.

The Yale face dataset was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

The CMU PIE face dataset contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. We use all the frontal faces with different illuminations and expressions in this test.

In these datasets, each face image is normalized to the resolution of 64×64 pixels based on the eye locations. The color images are converted to the grayscale with 256 levels per pixel and the histogram equalization is applied to all the images to reduce the disturbance of uneven illuminations.

For recognition, we randomly select half of faces from one subject as training samples and the other half are for test. For computational efficiency, the face images are preprocessed by PCA. The implementations of the subspace based recognition algorithms are quite similar to the steps in Algorithm 1. For sparse representation, which is not a subspace-based-method, we strictly follow the procedures in [28] and use the SolveLasso solver in sparse lab[5] for ℓ_1-norm minimization. For KNN graph construction, the value k is fixed by $n_t - 1$ where n_t is the average number of training samples for one subject. We show the recognition rate of manifold based algorithms on two graph topologies, i.e., KNN and sparse graph.

Table 1
Maximum recognition rates (%) on different datasets.

		Yale	PIE	AR	FERET
Linear subspace	PCA	87.4	95.5	62.2	63.6
	LDA	90.2	98.3	61.1	65.1
	NMF	85.5	95.8	63.2	64.2
Sparse	SR	**94.3**	93.3	76.8	73.1
KNN graph	LPP	90.7	98.1	74.5	66.5
	GEO	91.5	96.3	72.3	72.1
	CTG	93.5	**99.2**	80.2	78.3
Sparse graph	LPP	91.2	97.7	75.2	71.2
	GEO	90.1	98.1	78.2	79.3
	CTG	93.7	98.6	**81.2**	**79.4**

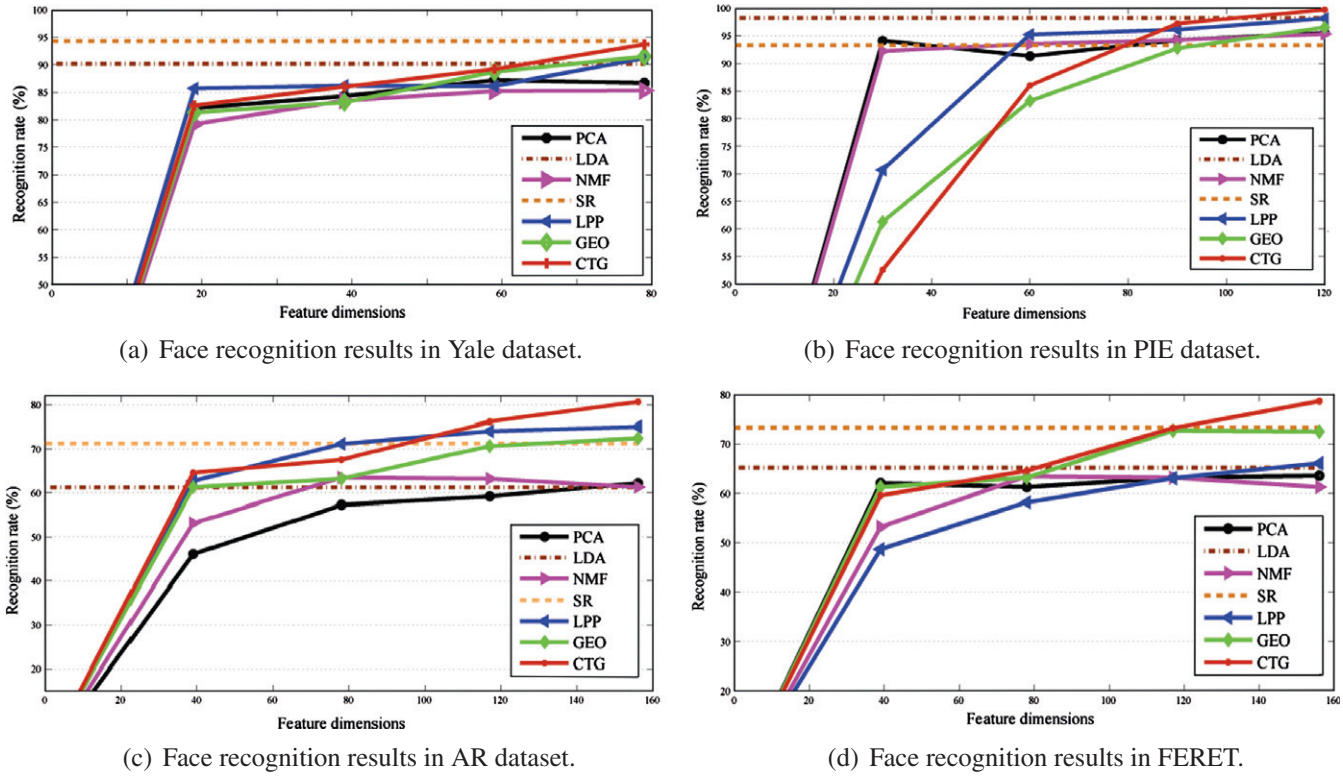
In the experiment, both the training and testing procedures are repeated for 10 times and the average recognition rates are reported. The maximal recognition rate of each method on four datasets are tabulated in Table 1 and their corresponding ROCs with different numbers of projection vectors are shown in Fig. 4.[6] In the table, the highest recognition rates are marked with bold letters. The results show that our random walk based method can achieve better performance levels than the other algorithms, in general. We will discuss the results on four datasets, respectively.

In Yale dataset, there are no significant differences among all the recognition methods. Generally speaking, the manifold based algorithms outperform PCA and NMF. Among the manifold based algorithms, CTG outperforms the other two. However, in Yale dataset, the best recognition rate is achieved by sparse representation (94.3%) which owns 0.6% improvements to the CTG on the sparse graph and has 0.8% improvements to the CTG on the KNN graph. In the PIE dataset, CTG on the KNN graph achieves the best perfor-

[4] The computer for implementing this algorithm is config with a Dual core 2.4 GHz CPU and a 4G RAM. The programs are operated on MATLAB 2008.

[5] http://sparselab.stanford.edu/

[6] For the manifold learning algorithms, we just show the ROCs on KNN graph in these figures.



(a) Face recognition results in Yale dataset.

(b) Face recognition results in PIE dataset.

(c) Face recognition results in AR dataset.

(d) Face recognition results in FERET.

Fig. 4. Recognition rate versus different feature dimensionality based on the four different datasets.

mance. SR loses its effectiveness since there are many faces with slightly pose variations. The basic assumption made in the SR is that it could only recognize frontal faces [28].

In Yale and PIE datasets, the improvements of CTG to other methods are not that significant. These two datasets are quite simple, on which typical algorithms can already achieve good performances. But, with the complicated and large datasets e.g. AR and FERET datasets, the manifold distribution of data will come out.

In the AR and the FERET datasets, three manifold based methods (LPP, GEO and CTG) significantly outperform linear subspace methods (PCA, LDA and NMF). Averagely, the manifold based approaches make about 15% improvements to the linear subspace methods. Moreover, CTG outperforms the other two manifold based methods on both the KNN graph and the sparse graph.

It is also interesting to note that the proposed CTG method is robust to graph topologies. The recognition performances by CTG are consistent on different graphs. However, the geodesic embedding (GEO) is much sensitive to graph topology. It achieves better performances on sparse graph while performs poorly one KNN graph.

4.4. Robustness verification

In previous parts, the experimental results on benchmark datasets demonstrate the effectiveness of the proposed CTG for general verification. In this part, we will further extend discussions to investigate the robustness of the CTGfaces to noises. The noise on faces always means illuminations and occlusions. Accordingly, in this part, two experiments on face recognition with illuminations and occlusions will be conducted. In order to avoid the heavy computational cost of the ℓ_1 minimization, all the graph based algorithms are performed on the KNN graph.

4.4.1. Face recognition with illumination

In this part, we will use the CTG-features to recognize faces captured under various illumination conditions. For this purpose, the extended Yale-B [31] is used.

The extended Yale-B dataset consists of 2414 frontal-face images of 38 individuals [31]. Each image is converted to grayscale and normalized to a size of 192×168. The histogram equalizations are applied to all the images. It worths noting that all the faces in Yale-B dataset are captured under various laboratory-controlled lighting conditions. Therefore, the extended Yale-B dataset is a desirable dataset for illumination test. Some faces captured under different lighting conditions in Yale-B are shown in Fig. 5.

There are 38 subjects in the extended Yale-B dataset. For each subject, we randomly select half of the images for training (about 32 faces per person) and the other half are for test. We will compare the CTG methods with PCA, LPP, GEO and SR. The training and testing procedures are repeated for 10 times. The average results and the ROCs are shown in Table 2 and Fig. 6, respectively.

The recognition results demonstrate the robustness of the CTG-faces for face recognition with illumination variations. The manifold based methods (CTG, GEO and LPP) outperform linear method, i.e., PCA. These findings also serve to back up the claims of some previous works, e.g. [9,1], that manifold-based-approaches are effective to recognize faces with illumination variations. Besides, on the same manifold, the CTG outperforms both LPP and GEO. The improvements may be ascribed to the robustness of commute time metric for manifold learning. The recognition rate of SR is about 4% lower than CTG in Yale-B dataset.

4.4.2. Occluded face recognition

Occluded face recognition is an important topic in computer vision [32]. The occlusions on the faces are always regarded as sparse noises [28]. In this part, we will test the CTG method to recognize the occluded faces in AR dataset.

Fig. 5. Some faces under different lighting conditions from the extended Yale-B dataset.

Table 2
The maximal recognition rates (%) with Illuminations (Yale-B).

Testing set	PCA	SR	LPP	GEO	CTG
Yale-B	67.8	73.5	71.1	76.2	**80.26**

Table 3
The maximal recognition rates (%) with occlusions (AR).

Testing sets	PCA	SR	LPP	GEO	CTG
AR1 (Sunglass)	52.2	**62.4**	56.2	58.3	60.9
AR2 (Scarf)	22.4	**49.3**	31.2	36.7	43.6

In AR dataset, for each subject, there are six occluded faces. Three faces are occluded by sunglasses and three are corrupted by scarfs. Some of these occluded faces are shown in Fig. 7. We divide them as two sub-sets which are denoted as AR1 (for occluded faces with sunglasses) and AR2 (for occluded faces with scarfs), respectively.

In the previous face recognition experiments on AR dataset, the occluded faces are treated the same as the non-occluded faces during both the training and testing procedures. However, in order to highlight the CTG method for occluded face recognition, in this part we use the normal faces in AR dataset for training and the occluded faces in AR1 and AR2 are for test.

This task is much challenging since the algorithm should recognize occluded faces from the completed training samples. Since there are no random selection strategy in this training and testing procedures, we just perform the experiment once. The recognition results are provided in Table 3 and Fig. 8, respectively.

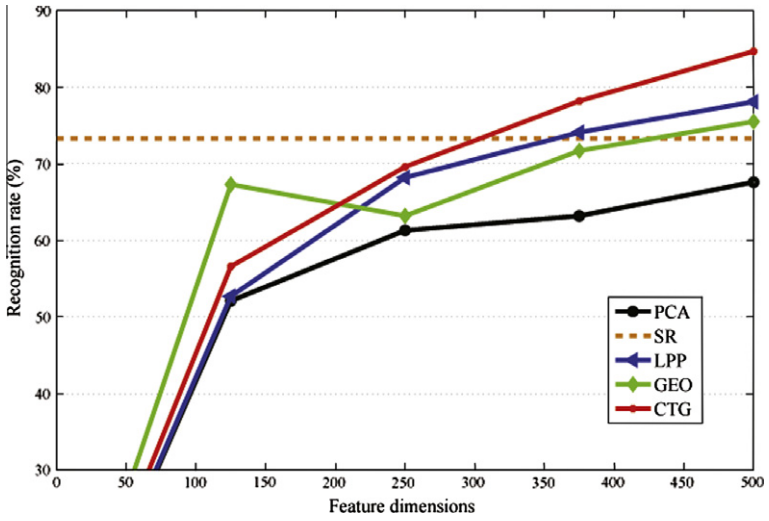
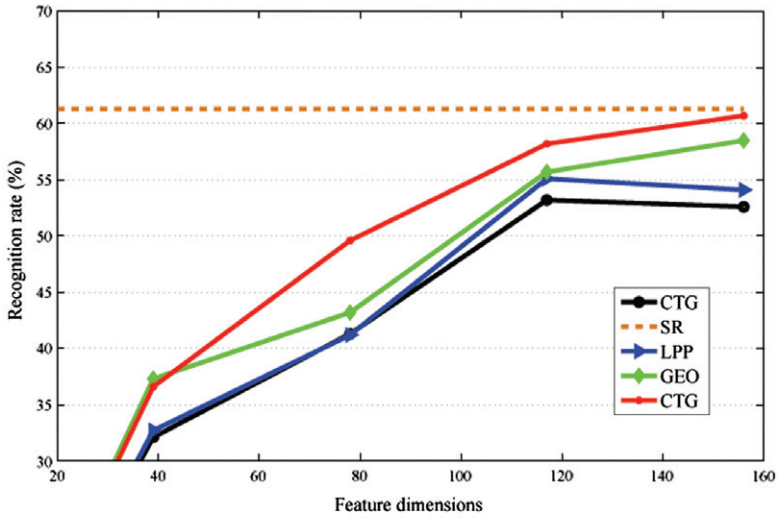
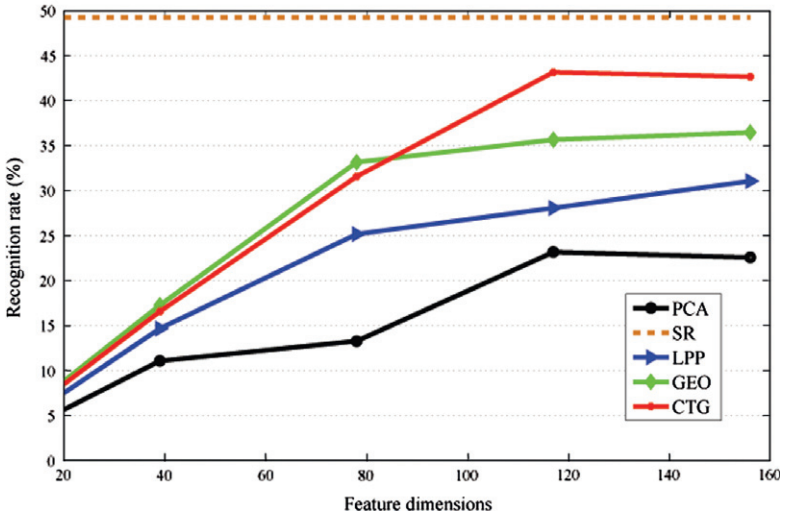


Fig. 6. Illumination test: face recognition rates with different feature dimensions in Yale-B dataset.

Fig. 7. The occluded faces in AR dataset.



(a) Occluded face recognition with sunglasses (AR1).



(b) Occluded face recognition with scarfs (AR2).

Fig. 8. Recognition rate versus different feature dimensionality based on the four different datasets.

From the results, it is concluded that sparse representation is the best one for occluded face recognition. The highest recognition rates on both AR1 and AR2 are achieved by the SR. SR is a robust method for occluded face recognition [25]. The recognition rates of other methods are all lower than SR. The drop is especially significant on the results in AR2, where SR makes nearly 27% improvements to PCA.

Among all the methods that are not based on sparse representation, the proposed CTG method achieves the highest recognition rate. On AR2, when recognizing faces with great occlusions (scarf), CTG achieves improvements as high as 6.9% to GEO and 10.6% to LPP owing to the robustness of commute time metric.

Compared with SR, there is one prominent advantage of the proposed CTG. The computational cost of SR is much heavier than CTG. When recognizing faces by SR, it requires to solve ℓ_1 optimization for all the testing samples. Averagely, it costs about 13 min to recognize all the faces in AR1 and costs 15 min in AR2. Nevertheless, the proposed CTG are much fast which can be finished in 2 min for all the faces in both AR1 and AR2.

5. Conclusions and discussions

In this paper, a graph-independent commute time embedding algorithm is proposed. This method generalize the learning result of commute time to the out-of-data samples. When applied to face recognition, our algorithm outperforms other typical methods on benchmark datasets. Besides, it is also efficient and effective to learn the faces that are disturbed by noise, e.g. illuminations and occlusions.

In summary, this paper has proposed a direction for feature extraction based on a random walk. However, the appeal is not limited to what is discussed in this paper. For example, in this paper, we have only focused our interest on image-domain-based features. It is believed that the proposed CTG model and the random-walk-based metric are also suitable for transformed-domain-based feature extraction. Some transformed-domain-based features, e.g. garbor features, are powerful tools to cope with occlusions and illuminations. It may be our future destination to combine the CTG methods with the garbor features to further

improve the performances of CTGfaces for occluded face recognition.

Acknowledgments

We greatly appreciate Prof. Yi Ma of UIUC and Dr. Allen Yang in U.C. Berkeley for their constructive suggestions on sparse representation. This work was supported by the National Basic Research Project (No. 2010CB731800), the Key Project of NSFC (No. 61035002) and the Science Fund for Creative Research Groups of NSFC (No. 60721003). Yue Deng was partially supported by the fellowship of Microsoft Research Asia, 2010.

Appendix A. Derivations of lagrangian minimization

In (11), $\langle\cdot\rangle$ is an inner product and we know that for two matrices, $\langle \mathbf{P},\mathbf{Q}\rangle = tr(\mathbf{P}^T\mathbf{Q}) = tr(\mathbf{Q}^T\mathbf{P})$. Therefore, for the sake of computational simplicity, we replace the second term in (11) by its trace and get:

$$L(\Lambda,\Omega) = tr(\Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T\Omega) - tr(\Omega^T\mathbf{N}\mathbf{A}\mathbf{N}^T\Omega\Lambda^T) + tr(\Lambda^T) \tag{A.1}$$

In order to get the optimal solution to Ω, we set $\nabla_\Omega L(\Omega,\Lambda) = 0$. The derivations of the two trace terms in (A.1) seems to be complicated. But, it can be greatly simplified by a general law of trace derivation. For any three matrices, **P**, **Q** and **W**,

$$\frac{\partial tr(\mathbf{W}^T\mathbf{P}\mathbf{W}\mathbf{Q})}{\partial \mathbf{W}} = \mathbf{Q}\mathbf{W}^T\mathbf{P} + \mathbf{Q}^T\mathbf{W}^T\mathbf{P}^T \tag{A.2}$$

Accordingly, we get:

$$\begin{aligned}\frac{\partial L}{\partial \Omega} &= \frac{\partial tr(\Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T\Omega)}{\partial\Omega} - \frac{\partial tr(\Omega^T\mathbf{N}\mathbf{A}\mathbf{N}^T\Omega\Lambda^T)}{\partial\Omega}\\ &= \Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T + \Omega^T[\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T]^T - \Lambda^T\Omega^T\mathbf{N}\mathbf{A}\mathbf{N}^T\\ &\quad - \Lambda^T\Omega^T(\mathbf{N}\mathbf{A}\mathbf{N}^T)^T\\ &= 2\Omega^T\mathbf{N}(\mathbf{A}-\mathbf{G})\mathbf{N}^T - 2\Lambda^T\Omega^T\mathbf{N}\mathbf{A}\mathbf{N}^T\end{aligned} \tag{A.3}$$

The third equality in (A.3) holds because all the matrices Λ, **A** and **G** are symmetric. We set (A.3) to be zero and get the generalized eigen-value decomposition equation in (12).

References

[1] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51, doi:10.1109/TPAMI.2007.250598.
[2] A. Martinez, A. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233, doi:10.1109/34.908974.
[3] K. Etemad, R. Chellappa, Discriminant analysis for recognition of human face images, J. Opt. Soc. Am. A 14 (1997) 1724–1733.
[4] D. Guillamet, J. Vitri, Non-negative matrix factorization for face recognition, in: M. Escrig, F. Toledo, E. Golobardes (Eds.), Topics in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2504, Springer, Berlin/Heidelberg, 2002, pp. 336–344.
[5] L.K. Saul, S.T. Roweis, Y. Singer, Think globally, fit locally: unsupervised learning of low dimensional manifolds, J. Mach. Learn. Res. 4 (2003) 119–155.
[6] J.B. Tenenbaum, V. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[7] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, doi:10.1126/science.290.5500.2323.
[8] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.
[9] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340, doi:10.1109/TPAMI.2005.55.
[10] R. Wilson, E. Hancock, E. Pekalska, R. Duin, Spherical embeddings for non-euclidean dissimilarities, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1903–1910. doi:doi:10.1109/CVPR.2010.5539863.
[11] H. Qiu, E. Hancock, Clustering and embedding using commute times, IEEE Trans. Pattern Anal. Mach. Intell. 29 (11) (2007) 1873–1890, doi:10.1109/TPAMI.2007.1103.
[12] M. Saerens, F. Fouss, L. Yen, P. Dupont, The principal components analysis of a graph, and its relationships to spectral clustering, Eur. Conf. Mach. Learn. 3201 (2004) 371–383.
[13] L. Hagen, A. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. 11 (9) (1992) 1074–1085, doi:10.1109/43.159993.
[14] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905, doi:10.1109/34.868688.
[15] H. Qiu, E.R. Hancock, Image segmentation using commute times, in: British Machine Vision Conference, 2005, pp. 929–938.
[16] R. Behmo, N. Paragios, V. Prinet, Graph commute times for image representation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[17] Y. Deng, Q. Dai, Z. Zhang, Feature extraction using randomwalks, in: YC-ICT '09, 2009: IEEE Youth Conference on Information, Computing and Telecommunication, 2009, pp. 498–501. doi:doi:10.1109/YCICT.2009.5382449.
[18] F. Gobel, A. Jagers, Random walks on graphs, Stoch. Proces. Appl. 2 (1974) 311–336.
[19] F. Fouss, A. Pirotte, M. Saerens, A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation, in: IEEE/WIC/ACM International Conference on Web Intelligence, 2005, pp. 550–556.
[20] J. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27, doi:10.1007/BF02289565. <http://dx.doi.org/10.1007/BF02289565>.
[21] J. Kruskal, Nonmetric multidimensional scaling: a numerical method, Psychometrika 29 (1964) 115–129, doi:10.1007/BF02289694. <http://dx.doi.org/10.1007/BF02289694>.
[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2010) 1–123.
[23] A. Martinez, R. Benavente, The AR Face Database, Computer Vision Center, CVC Technical Report.
[24] B. Cheng, J. Yang, S. Yan, T. Huang, Learning with l1 graph for image analysis, IEEE Trans. Image Proces. 19 (4) (2010) 858–866.
[25] D. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306, doi:10.1109/TIT.2006.871582.
[26] H. Cheng, Z. Liu, J. Yang, Sparsity induced similarity measure for label propagation, in: International Conference on Computer Vision, 2009, pp. 317–324, doi:doi:10.1109/ICCV.2009.5459267.
[27] P. Phillips, H. Moon, S. Rizvi, P. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1090–1104, doi:10.1109/34.879790.
[28] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227, doi:10.1109/TPAMI.2008.79.
[29] P. Belhumeur, D. Kriegman, The yale face database <http://cvc.yale.edu/projects/yalefaces/yalefaces>.
[30] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (pie) database of human faces <http://www.ri.cmu.edu/pubs/download>.
[31] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660, doi:10.1109/34.927464.
[32] Y. Deng, Q. Dai, Z. Zhang, Graph laplace for partially occluded face completion and recognition, IEEE Trans. Image Proces. 20 (8) (2011) 2329–2338.