Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

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

Transformation invariant subspace clustering

Qi Li [a], Zhenan Sun [a,*], Zhouchen Lin [b,c], Ran He [a,*], Tieniu Tan [a]

[a] Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China
[b] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, China
[c] Cooperative Medianet Innovation Center, Shanghai Jiaotong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 14 August 2015
Received in revised form
2 February 2016
Accepted 4 February 2016

Keywords:
Transformation
Subspace clustering
Joint alignment and clustering

ABSTRACT

Subspace clustering has achieved great success in many computer vision applications. However, most subspace clustering algorithms require well aligned data samples, which is often not straightforward to achieve. This paper proposes a Transformation Invariant Subspace Clustering framework by jointly aligning data samples and learning subspace representation. By alignment, the transformed data samples become highly correlated and a better affinity matrix can be obtained. The joint problem can be reduced to a sequence of Least Squares Regression problems, which can be efficiently solved. We verify the effectiveness of the proposed method with extensive experiments on unaligned real data, demonstrating its higher clustering accuracy than the state-of-the-art subspace clustering and transformation invariant clustering algorithms.

1. Introduction

Clustering is a core research problem with numerous applications in computer vision and pattern recognition. It aims to group a set of unlabeled data according to some intrinsic structure of data. Among many clustering algorithms, subspace clustering has been extensively studied in recent years [1–9,5]. It refers to recovering the underlying subspaces and determining the membership of each data sample to the subspaces. It has been widely used in handwritten digits clustering [4,10,11], face clustering [1,2,4,12], motion segmentation [1,2,4,12–14], etc. Compared with traditional clustering algorithms, subspace clustering is better formulated and is more robust to noise, sparse outliers, and missing entries.

Subspace clustering assumes that data samples are drawn from some linear subspaces. This assumption is reasonable and useful for many kinds of data, such as human faces captured under varying illumination [15] and feature trajectories of a moving rigid object in a video [16]. In the past decade, many subspace clustering algorithms have been proposed to recover the linear subspaces that data samples reside in and have achieved superior clustering results in many applications. However, existing subspace clustering algorithms require well aligned data samples. This may not always be true for real-world data, e.g., face images downloaded from social network websites. So these subspace clustering algorithms may fail to find the ground-truth subspace structures due to misalignment.

Recently, clustering misaligned data samples has drawn considerable attention from researchers [17–19]. In [18], transformation and clustering parameters are formulated into a unified objective function which is defined as weighted l_2 distances between image pairs. The objective function consists of two parts: the within-cluster difference and the between-cluster difference. The within-cluster difference is defined as:

$$\varepsilon_{int} = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \pi_i^T \pi_j \|A_{ij}\|^2, \tag{1}$$

where n is the total number of misaligned images, $\pi_i = [\pi_{i,1}, \ldots, \pi_{i,c}]$ is a c-dimensional vector which represents the probability that i-th image belongs to the c-th cluster, A_{ij} is the difference of the i-th image and j-th image. The between-cluster image difference is represented as:

$$\varepsilon_{ext} = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \alpha_{ij} \|A_{ij}\|^2, \tag{2}$$

where $\alpha_{ij} = 1 - \pi_i^T \pi_j$. By the motivations of driving the images of the same cluster to be close and encouraging the images of different clusters to be far, Eqs. (1) and (2) are combined to form a unified objective function:

$$\varepsilon_{total} = \varepsilon_{int} - \lambda \varepsilon_{ext}. \tag{3}$$

where λ is used to balance the two terms. Then an iterative optimization method is proposed to solve Eq. (3) to get the alignment

* Corresponding authors.
E-mail addresses: qli@nlpr.ia.ac.cn (Q. Li), znsun@nlpr.ia.ac.cn (Z. Sun), zlin@pku.edu.cn (Z. Lin), rhe@nlpr.ia.ac.cn (R. He), tnt@nlpr.ia.ac.cn (T. Tan).

http://dx.doi.org/10.1016/j.patcog.2016.02.006
0031-3203/

parameters and cluster labels. A transformed Bayesian infinite mixture model with a Dirichlet process prior that can simultaneously align and cluster a data set is proposed by Mattar et al. [19]. They describe two different learning schemes to learn the transformation and the cluster parameters. Experiments show that joint alignment and clustering offers many advantages over the traditional approach which clusters the data set first and then aligns the data set. In [17], the transformation invariant clustering (TIC) algorithm applies a mixture of Gaussians to include data transformation as a latent variable, producing a transformation invariant Gaussian mixture model for clustering data misaligned due to local and global transformations. One drawback of TIC is that it can only be used with a discrete set of allowable spatial transformations. Although these three clustering methods partially alleviate the misalignment problem, they still need additional techniques, e.g., applying histogram of oriented gradients (HOG), to improve clustering accuracy.

Inspired by the observations in [18,19] that alignment and clustering are two highly coupled problems, and solving both of them together could offer many advantages over clustering and aligning images separately, this paper integrates aligning data samples and learning their subspace representation into a joint framework we term Transformation Invariant Subspace Clustering (TISC). Our framework simultaneously seeks an optimal set of transformations that align data samples and a linear representation of subspace structures that produces an affinity matrix among samples. The joint optimization problem can be reduced to a sequence of Least Squares Regression problems, which can be efficiently solved. In addition, we show that the Least Squares Congealing algorithm [22] is a special case of our framework. Experimental results on various databases show that our framework outperforms simple combination of alignment and subspace clustering. Compared with previous transformation invariant clustering algorithms [18,19] that use specific feature representations (e.g., HOG) to improve accuracy, our proposed framework can surpass state-of-the-art results even working on raw data. Fig. 1 shows a diagram of our algorithm.

The main contributions are summarized as follows:

(1) A transformation invariant subspace clustering algorithm is proposed to alleviate the misalignment problem in subspace clustering. To the best of our knowledge this is one of the first algorithms to incorporate transformation invariance into subspace segmentation. Compared with previous transformation invariant clustering algorithms [18,19] that use specific feature representations (e.g., HOG) to improve accuracy, the proposed method can achieve state-of-the-art results even based on image pixels.
(2) Experimental results on various databases have shown that our algorithm performs better than the simple combination of image alignment and subspace clustering.

The rest of this paper is organized as follows. In Section 2, we review some recent advances in subspace clustering that involve image alignment. Then we present the technical details on our TISC framework in Section 3. We show the experimental results in Section 4. Finally we conclude our paper in Section 5.

2. Related work

In this section, we briefly introduce the basic concept of subspace clustering and review recent advances in subspace clustering that involve image alignment.

Given a set of data samples drawn from a union of multiple subspaces, the goal of subspace clustering is to segment data samples into clusters with each cluster corresponding to a subspace. There has been a lot of work on subspace clustering. These methods can be roughly divided into four main categories: algebraic methods [23–25], iterative methods [26–28], statistical methods [29–31], and spectral-clustering-based methods [1,2,10,3,32,33,9]. Among them, the spectral-clustering-based methods have shown excellent performance in many real applications.

The representative spectral-clustering-based subspace clustering algorithms include the sparse subspace clustering (SSC) [1,34], Least Squares Regression (LSR) [3], low-rank representation (LRR) [2,12], correlation adaptive subspace segmentation (CASS) [10], discriminative subspace clustering (DiSC) [4], greedy subspace clustering (GSC) [14], smooth representation clustering (SMR) [11],

Fig. 1. *A diagram of our algorithm.* The images are selected from the Labeled Faces in the Wild (LFW) database [20]. The faces in the left are obtained by the Viola–Jones face detector [21]. The faces in the right are clustered and aligned by our algorithm with each row representing one estimated cluster.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

robust subspace segmentation with block-diagonal prior (RSS-BD) [8], robust subspace clustering via thresholding (RSCT) [35], structured sparse subspace clustering (SSSC) [9], etc. SSC, LSR, LRR, CASS, SMR, RSS-BD and SSSC learn an affinity matrix whose entries measure the similarities among the data samples and then, based on the learned affinity matrix, use spectral clustering methods to segment data samples. GSC uses the nearest subspace neighbor method to construct a neighborhood matrix, and then recovers subspaces in a greedy fashion. RSCT resorts to Thresholding Ridge Regression to project the data set into the linear projection space, and normalized spectral clustering is used to segment data samples. DiSC solves the clustering problem by using a quadratic classifier trained from the unlabeled data samples (clustering by classification).

A key step of the subspace clustering algorithms is to find the correlation among data samples to construct an affinity matrix. The underlying idea of SSC is that each data sample in a union of subspaces can be effectively represented as a linear combination of other samples from its own subspace, which is called the "self-expressiveness" property of data samples [36–39]. To increase computational efficiency and also ensure high clustering accuracy, LSR resorts to Least Squares Regression to model the correlation among data samples. LRR aims to find the lowest-rank representation among all data samples. CASS applies trace Lasso which is adaptive to data correlation. It can be regarded as a tradeoff between SSC and LSR. SMR enforces the grouping effect explicitly by the affinity of samples on the regularization term. RSS-BD introduces the Laplacian constraint for block-diagonal matrix pursuit. There have been many variations of SSC, LSR, LRR, CASS, SMR and RSS-BD. One limitation of these subspace clustering algorithms is that they cannot be extended to nonlinearly distributed data samples. In fact, they may fail in many real-world applications, such as clustering the handwritten digits automatically or clustering face images in surveillance video sequences.

For many kinds of data, such as images and videos, the nonlinearity of data distribution is due to misalignment among data samples [40,20,41–46]. So if we could resolve the misalignment issue, the traditional subspace clustering algorithms can still be applied. However, few researchers have addressed the issue of misalignment in subspace clustering. Recently, some algorithms have been proposed to solve the misalignment problem, such as robust alignment by sparse and low-rank decomposition [47,48], transformation invariant low-rank textures [49], robust alignment by sparse representation [37,50], simultaneous image transformation and sparse representation recovery [51], coupling alignments and recognition [52], and transformation invariant PCA [53]. These recent advances that involve image alignment motivate us to develop a new transformation invariant method for subspace clustering, in order to successfully recover the subspaces of misaligned data samples.

3. Transformation invariant clustering algorithm

In this section, we begin with a new cost function that measures image similarity by finding the linear subspaces of misaligned data. Then a transformation invariant clustering algorithm is proposed along with the discussion about previous algorithms. Moreover, we analyze the convergence rate and time complexity of our algorithm. Finally, we demonstrate the relationship between the well-known Least Squares Congealing algorithm [22] and our algorithm.

3.1. Problem statement

Transformation invariant clustering algorithms aim to simultaneously learn background cluster labels and spatial transformations from an unlabeled complex database. They are often motivated by two observations on misaligned data samples: (1) if aligned data samples span linear subspaces, their moderately unaligned versions lie on a union of nonlinear manifolds, which can still be mapped to linear subspaces in a higher dimensional ambient space, and (2) if data samples live on the nonlinear manifolds, by estimating the transformations accurately the nonlinear manifolds can be rectified into linear subspaces.

By the first observation, we can convert nonlinear manifolds into linear subspaces by embedding them in a higher dimensional space. However, increasing the dimension of nonlinear manifolds has the drawback of possibly increasing the dimension of the intersection of two subspaces, which may result in indistinguishability between subspaces [1]. In addition, it will no longer be effective when the misalignment is large. So we turn to the second observation. That is to estimate transformations as accurately as possible. Since we need to know the parameterization of the transformations, in this paper we focus on aligning image data. Nonetheless, in theory there is no restriction on applying TISC to other data as long as we know the parameterization of transformation.

Before discussing the joint learning problem, we first introduce the basic technique of subspace clustering. For n well aligned images, we denote the operator $\mathbb{R}^{w\times h}\to\mathbb{R}^m$ as selecting an m-pixel region of interest from the i-th input image ($i=1,\ldots,n$), and stack it as a column vector represented by $I_i\in\mathbb{R}^m$, then the n well aligned images can be represented as a matrix $D=[I_1,\ldots,I_n]\in\mathbb{R}^{m\times n}$. A natural assumption on these well aligned images is that they are linearly correlated to each other [47,48,54]. Hence they lie in a union of multiple subspaces. We can summarize the models of several spectral-clustering-based subspace clustering algorithms as follows:

$$\min_Z \|D-DZ\|_p+\lambda\|Z\|_q,\quad \text{s.t.}\quad \operatorname{diag}(Z)=0, \tag{4}$$

where $Z\in\mathbb{R}^{n\times n}$ is the affinity matrix with $|Z_{ij}|$ measuring the similarity between the i-th and j-th aligned images, $\lambda>0$ is a parameter used to balance the above two terms, and p and q is a matrix norm, such as the ℓ_0-norm, ℓ_1-norm, nuclear norm, and Frobenius norm.[1]

Due to change of positions or poses in practical applications, Eq. (4) may not be applicable for misaligned images. Hence, we further introduce aligning parameters $\tau=[\tau_1,\cdots,\tau_n]$ for all the n misaligned images, where τ_i is a p-dimensional vector that aligns each image to a predefined common coordinate space with respect to a specific group of transformations. Let vector x be a collection of m-pixel region of interest defined in the common space. $W(x;\tau_i)$ is the aligning function that maps the coordinate of the original input image I_i to the coordinate of the common space. Then we can model the aligned image vector of m-pixel region of interest from each input image $I_i(x)$ as $I_i(W(x;\tau_i))\in\mathbb{R}^m$. In the sequel, we use $I_i\circ\tau_i$ to represent $I_i(W(x;\tau_i))$ for convenience. Then the n aligned images can be represented as $D\circ\tau=[I_1\circ\tau_1,\cdots,I_n\circ\tau_n]\in\mathbb{R}^{m\times n}$. As a result, we obtain a general problem for joint alignment and subspace clustering as follows:

$$\min_{\tau,Z}\ \|D\circ\tau-(D\circ\tau)Z\|_p+\lambda\|Z\|_q,\quad \text{s.t.}\quad \operatorname{diag}(Z)=0. \tag{5}$$

The first term in Eq. (5) represents the errors to recover linear subspaces of aligned images. The second term in Eq. (5) is a regularization on the affinity matrix.

[1] Given a matrix $A\in\mathbb{R}^{m\times n}$, the ℓ_0-norm of A is defined as the number of nonzero elements in A. The ℓ_1-norm of A is defined as $\|A\|_1=\sum_{i=1}^m\sum_{j=1}^n|A_{ij}|$. The nuclear norm of A is defined as $\|A\|_*=\sum_{i=1}^{\min\{m,n\}}\sigma_i(A)$, where $\{\sigma_i(A)\}$ are the singular values of A. The Frobenius norm of A is defined as $\|A\|_F=\left(\sum_{i=1}^m\sum_{j=1}^n A_{ij}^2\right)^{1/2}$.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

There are many variations of Eq. (5) if different matrix norms are adopted. In this paper, we only use the simplest squared Frobenius norm in Eq. (5), resulting in the following optimization problem:

$$\min_{\tau,Z} \quad \|D\circ\tau-(D\circ\tau)Z\|_F^2+\lambda\|Z\|_F^2, \quad \text{s.t.} \quad \mathrm{diag}(Z)=0. \tag{6}$$

3.2. Solving Eq. (6)

Problem (6) is difficult to solve because $D\circ\tau$ depends on the transformations parameterized by τ. We solve it by alternating minimization, or called block coordinate descent. Namely, we update Z by fixing τ at the k-th iteration:

$$Z^{k+1}=\arg\min_{Z}\|D\circ\tau^k-(D\circ\tau^k)Z\|_F^2+\lambda\|Z\|_F^2, \quad \text{s.t.} \quad \mathrm{diag}(Z)=0. \tag{7}$$

Then we update τ based on the obtained Z:

$$\tau^{k+1}=\arg\min_{\tau}\|D\circ\tau-(D\circ\tau)Z^{k+1}\|_F^2. \tag{8}$$

When the transformations are differentiable, the above iteration converges to a critical point of Eq. (6).

Eq. (7) is a constrained least squares problem. By introducing a Lagrange multiplier y, we have the Lagrangian function of Eq. (7):

$$L(Z,y)=\|D_k-D_kZ\|_F^2+\lambda\|Z\|_F^2+\langle y,\mathrm{diag}(Z)\rangle, \tag{9}$$

where $D_k=D\circ\tau^k$. The optimal Z is a saddle point of L:

$$\frac{\partial L}{\partial Z}=-2D_k^T(D_k-D_kZ)+2\lambda Z+\mathrm{diag}^*(y)=0, \tag{10}$$

where $\mathrm{diag}^*(y)$ is a diagonal matrix whose diagonal entries are y and diag^* is the adjoint operator of diag. The solution to Eq. (10) is:

$$Z=\left(D_k^TD_k+\lambda I\right)^{-1}\left(D_k^TD_k-\tfrac{1}{2}\mathrm{diag}^*(y)\right). \tag{11}$$

The choice of y should make Z fulfill the constraint $\mathrm{diag}(Z)=0$. It is determined as follows. We reformulate the solution as:

$$\left(D_k^TD_k+\lambda I\right)^{-1}\left[\left(D_k^TD_k+\lambda I\right)-\left(\lambda I+\tfrac{1}{2}\mathrm{diag}^*(y)\right)\right] =I-\left(D_k^TD_k+\lambda I\right)^{-1}\left(\lambda I+\tfrac{1}{2}\mathrm{diag}^*(y)\right)=I-\left(D_k^TD_k+\lambda I\right)^{-1}\Lambda, \tag{12}$$

where $\Lambda=\lambda I+\frac{1}{2}\mathrm{diag}^*(y)$, Λ is a diagonal matrix. If we can decide Λ, then y can be decided as:

$$\mathrm{diag}^*(y)=2(\Lambda-\lambda I). \tag{13}$$

Since Λ is diagonal, $\left(D_k^TD_k+\lambda I\right)^{-1}\Lambda$ in Eq. (12) is simply scaling the columns of $\left(D_k^TD_k+\lambda I\right)^{-1}$. So the diagonal entries of $\left(D_k^TD_k+\lambda I\right)^{-1}\Lambda$ is $\left[\left(D_k^TD_k+\lambda I\right)^{-1}\right]_{ii}\Lambda_{ii}$. Therefore to make the diagonal entries of Z zeros, Λ must be chosen as:

$$\Lambda_{ii}=\frac{1}{\left[\left(D_k^TD_k+\lambda I\right)^{-1}\right]_{ii}}. \tag{14}$$

Accordingly, the optimal Z and Λ are given by Eqs. (12) and (14).

Given Z^k, we then employ an iterative optimization method to solve the nonlinear problem (8) [55,22,48,49,56]. As in RASL and TILT, we minimize the problem iteratively by local linearization, i.e.,

$$I_i\circ\tau_i\approx I_i^r+J_i\Delta\tau_i, \tag{15}$$

where $I_i^r=I_i\circ\tau_i^r$, τ_i^r is the value of τ_i at the r-th iteration to solve (8), $J_i=\frac{\partial}{\partial\xi}(I_i\circ\xi)\,|_{\,\xi=\tau_i}\in\mathbb{R}^{m\times p}$ is the Jacobian of the i-th image at the parameters τ_i^r, and $\Delta\tau_i\in\mathbb{R}^{p\times 1}$ is the increment of the currently estimated parameter τ_i^r. J_i can be calculated as $J_i=\nabla I_i(\partial W(x;\tau_i)/\partial\tau_i)$, where ∇I_i is the gradient of image I_i evaluated at $W(x;\tau_i)$, $(\partial W(x;\tau_i)/\partial\tau_i)$ is determined by the specific transformation type. Then we solve

$$\min_{\Delta\tau}\|X^r-X^rZ\|_F^2, \tag{16}$$

where $X^r=[I_1^r+J_1\Delta\tau_1\cdots I_n^r+J_n\Delta\tau_n]$ and we drop the subscript k of Z for simplicity. Given the solution to Eq. (16), we update τ_i as follows,

$$\tau_i^{r+1}=\tau_i^r+\Delta\tau_i. \tag{17}$$

The above iteration goes until convergence.

Algorithm 1. Transformation Invariant Subspace Clustering (TISC).

Input:
The set of the misaligned images $I_1,\ldots,I_n$, Initial transformations $\tau_1,\ldots,\tau_n$.

1: **while** not converged **do**

2: Compute the Jacobian matrix with respect to the transformation parameter:

$$J_i=\frac{\partial}{\partial\xi}\left(\frac{I_i\circ\xi}{\|I_i\circ\xi\|_2}\right)\Bigg|_{\xi=\tau_i}, \quad i=1,\ldots,n.$$

3: Normalize the aligned images:

$$\frac{I_i\circ\tau_i}{\|I_i\circ\tau_i\|_2}, \quad i=1,\ldots,n.$$

4: Compute: $D_k=D\circ\tau$, $Z=I-\left(D_k^TD_k+\lambda I\right)^{-1}\Lambda$.

5: **for** $i=1{:}n$ **do**

6: $\Delta\tau_i=J_i^+(D_kZ_i-I_i\circ\tau_i)$;

7: **end for**

8: Update the parameter: $\tau_i=\tau_i+\Delta\tau_i$, $i=1,\ldots,n$.

9: **end while**

Output:
The final alignment parameter $\tau=[\tau_1,\ldots,\tau_n]$ and affinity matrix $Z=[Z_1,\ldots,Z_n]$.

Eq. (16) is a standard least squares problem. We can get its analytic solution by taking its partial derivative (details are presented in the appendix). However, the analytic solution of Eq. (16) requires solving a large linear system, resulting in high computational cost in each iteration. Hence we follow the way of the classic Lucas-Kanade algorithm [55] and make use of an approximate way provided in [22,57]. The strategy in [22,57] involves successively aligning a single image I_i with the rest of the ensemble. During the iteration, the transformation parameters of the rest of the ensemble are fixed. The set of parameters are then obtained by sequentially aligning and then updating each image in the stack until convergence. Experimental results in Section 4 demonstrate that this approximate way can also lead to state-of-the-art results.

By the above idea, we expand the matrix Z and rewrite Eq. (6) as follows:

$$\min_{\Delta\tau_i}\sum_{i=1}^{n}\left\|(I_i^r+J_i\Delta\tau_i)-\sum_{\substack{j=1\\ j\neq i}}^{n}\left(I_j^r+J_j\Delta\tau_j\right)Z_{ji}\right\|, \tag{18}$$

where Z_{ji} denotes the entry at the j-th row and i-th column of the matrix Z. Similar to Learned–Miller [57] and Cox et al. [22], we simply initialize $\Delta\tau_j$ as zeros when $j\neq i$. Then the solution can be

obtained as:

$$\Delta\tau_i = J_i^+ \left((D\circ\tau^r)Z_i - I_i^r\right), \tag{19}$$

where J_i^+ is the Moore-Penrose pseudoinverse of J_i and Z_i is the i-th column of Z.

Finally, we get the transformation parameter τ and the affinity matrix Z. Algorithm 1 summarizes the above minimization procedure. Given the affinity matrix Z, Normalized Cuts [58] is used to obtain the final clustering results.

3.3. Convergence and complexity

Replacing the nonlinear problem with a sequence of linear problems can be viewed as a Gauss–Newton method. It has been used in image alignment since the birth of Lucas–Kanade [59] algorithm. This approximation usually consists of two steps. The first step is to minimize the nonlinear optimization problem by making a linear approximation to get an increment of the parameters. Then update the current estimation of the parameters using the increment. Detailed analysis of convergence rate of Gauss–Newton method can be found in [55]. Our algorithm can also be treated as the Gauss–Newton method for minimizing a nonlinear problem.

Suppose that the alignment parameter τ_i is a p-dimensional vector, the number of pixels in an aligned image is m, and the total number of images is n. Then the computational cost of each loop of our algorithm is $O(p^2mn+n^3+p^3n)$. When the transformation type and the number of pixels in aligned images are fixed, the most expensive computation is on $\left(D_k^TD_k+\lambda I\right)^{-1}$, which is $O(n^3)$. Lots of methods can be used to reduce this cost when dealing with a large numbers of images. We leave this issue to future work. In fact, with our unoptimized algorithm and codes, in the following experiments our algorithm only uses a few minutes (less than 10) to cluster hundreds of misaligned images selected from the LFW database.

3.4. Relation to previous work

If there is no misalignment in data samples, we can remove the transformation parameter τ from Eq. (6). Then Least Squares Regression subspace clustering algorithm [3] becomes a special case of our general formulation in Eq. (6). The Least Squares Congealing algorithm [22] is also a special case of our algorithm. Its objective function takes the form [22],

$$\varepsilon_i(q) = \sum_{j=1, j\neq i}^{N} \left[I_j - I_i(q)\right]^2, \tag{20}$$

where q is the aligning parameter, ε_i is the cost function for the i-th image, $I_j(j\neq i)$ is the rest of the ensemble during each iteration. Eq. (20) is also a highly non-linear function and difficult to minimize directly. They linearize Eq. (20) by taking the first order Taylor series approximation around $I_i(q)$, resulting in the following problem:

$$\min_{\Delta q} \sum_{j=1, j\neq i}^{N} \left[I_j - I_i(q) - \frac{\partial I_i(q)^T}{\partial q}\Delta q\right]^2, \tag{21}$$

where $\frac{\partial I_i(q)}{\partial q}$ is the steepest descend image and Δq is the incremental variable which needs to be estimated. If we fix the parameter Z in TISC to be an all-one matrix except that its diagonal entries are zeros, then for the i-th image we can rewrite Eq. (18) as follows:

$$\min_{\Delta\tau} \sum_{j=1, j\neq i}^{n} \left(I_i\circ\tau_i + J_i\Delta\tau_i - I_j\right)^2. \tag{22}$$

Then Eqs. (21) and (22) become the same minimization problem. Compared with Least Squares Congealing, our algorithm can adaptively adjust Z for the aligned images according to their underlying subspaces, rather than merely estimate from the average of these aligned images.

4. Experiments

To evaluate TISC's performance, we conduct the experiments on a variety of databases. First, we test TISC on the Dummy head database [47] to see its ability to cope with varying levels of misalignment and occlusion. Then we conduct experiments on the MNIST database [40], the CMU Multi-PIE database [60], the PubFig database [61], and the LFW database [20] to evaluate the clustering performance of TISC against state-of-the-art clustering algorithms. Finally, we illustrate how crucial the regularization parameter can affect the clustering performance.

4.1. Implementation details

When our algorithm is used to cluster and align binary images, there exists a degenerate case called “drift”. For a set of nonrigid transformations, like the affine transformations, the proposed objective function can simply decimate binary images by shrinking all of them to a pixel. A technique for mitigating this shrinking problem is suggested by Learned-Miller [57]: to constrain each of the parameters to have zero mean across all of the image set. To incorporate this idea into our algorithm, we subtract the mean value of each parameter from the values of that parameter for each image periodically. Similar to other subspace clustering algorithms, the parameter λ of our algorithm is tuned empirically around 0.1. We can set the parameter in a greedy way to determine the best one. All experiments are implemented in MATLAB on a PC with an Intel i7-2600 3.40 GHZ CPU and 8 GB memory.

4.2. Alignment performance on the dummy head database

For this experiment, 100 images from the Dummy head database [48] taken under different illumination conditions are used to evaluate TISC's ability to cope with various levels of misalignment. We synthetically perturb a random Euclidean transformation on each of the images. The assignment is to align the perturbed image to an 80×60-pixel canonical frame where the distance between the outer eye corners is normalized to 50 pixels. The manually labeled coordinates of the outer eye corners are used as the reference points and the difference between the coordinates of the outer eye corners after alignment is used as a metric to assess if the alignment is successful or not.

Different levels of uniformly distributed Euclidean transformations are enforced on the input images. The x and y-translations are uniformly distributed in $[-x/2, x/2]$ and $[-y/2, y/2]$ respectively. The angles of rotation are uniformly distributed in $[-\theta/2, \theta/2]$. The metric of successful alignment is that the maximal difference in both x and y-coordinates of the outer eye corners across all pairs of images is less than one pixel in the canonical frame. Fig. 2 shows the successful rate over 10 independent trails. From Fig. 2, we can see that our algorithm can align the image successfully when x and y-translation are both less than 15 pixels or y-translation less than 10 pixels, and at the same time the angles of rotation less than 60 degrees. This experiment has shown that although the linearization step in our algorithm holds locally (the same as many other image alignment algorithms), it can correctly align the misaligned images in a large region of attraction.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

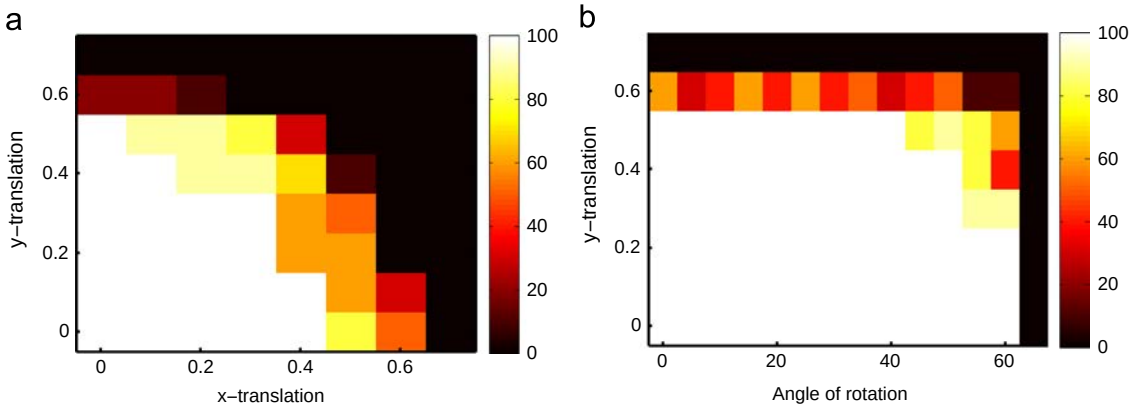

Fig. 2. (a) Simultaneous translation in the x and y directions. (b) Simultaneous translation in y direction and in-plane rotation. The x and y-translations are represented by the percent of the normalized pixels between the outer eye corners.

4.3. Results on the MNIST database

The MNIST database is the most commonly used dataset to evaluate different clustering algorithms, due to its diverse sample variation [18,19,4]. It has been considered one of the most difficult databases. As in [18,19], we select a total number of 200 images from this database.[2] Therefore, we can have both qualitative and quantitative comparison with the published results. We choose the affine transformation to cluster and align the handwritten digits and set the initial transformation as the identity transformation. The experimental results include visual comparison on the average digits before and after alignment, using the estimated cluster labels. We use two metrics suggested by [18,19] to evaluate the quantitative performance of different alignment and clustering algorithms. (1) Rand index, which evaluates the clustering accuracy with respect to the ground truth labels. (2) Alignment score, which measures the average of squared pair-wise distance between any two digits belonging to the same estimated cluster. The mean and standard deviation of all the distances are reported.

Table 1 summarizes the quantitative clustering and alignment results. The results by K-means, Congealing, and JAC are quoted from [19], while those of TIC, USAC, and SSAC are from [18]. In K-means, the K-means clustering algorithm is run 200 times and the best clustering result is chosen. Congealing refers to congealing on all of the images and then clustering the well aligned images using the K-means algorithm. Semi-supervised JAC refers to semi-supervised joint alignment and clustering using a Bayesian nonparametrics algorithm. It uses a single positive example for each digit. As for our algorithm, we do not use any information about the cluster labels. Unsupervised JAC means the unsupervised joint alignment and clustering algorithm. TIC represents using the online implementation of Transformation Invariant Clustering algorithm. USAC and SSAC denote the unsupervised and semi-supervised simultaneous clustering algorithms, respectively.

Table 1 shows that TISC achieves the highest clustering accuracy and the best alignment performance. It seems that the improvement of TISC over Unsupervised JAC is incremental. However, unsupervised JAC performs clustering on this database with 12 clusters (different from all of the other algorithms listed in Table 1), which makes it perform better than other algorithms except TISC. Given the same clustering assignment, the accuracy of unsupervised JAC must be worse than that of semi-supervised JAC which utilizes the ground truth label information. As shown in Table 1, TISC performs better than semi-supervised JAC by five percent, which indicates that TISC surpasses unsupervised JAC by at least the same margin.

Table 1
Alignment and clustering results on the MNIST database. All of the alignment scores are divided by 10^6.

Algorithm	Clustering (%)	Alignment
K-means	62.5	4.88 ± 1.6
Congealing [57]	70.5	3.51 ± 1.3
Semi-supervised JAC [19]	82.5	2.71 ± 1.3
Unsupervised JAC [19]	87.0	2.38 ± 1.1
TIC [17]	35.5	6.00 ± 1.1
USAC [18]	56.5	3.80 ± 0.9
SSAC [18]	73.7	–
TISC	**87.5**	$\mathbf{2.32 \pm 1.6}$

We further show the average digits obtained by different algorithms after alignment in Fig. 3. We observe that the average digits of TISC are clearer and cleaner than other algorithms, except the average of digit "1". The reason is that sometimes our algorithm confuses digit "1" with digit "3" during the clustering process. Such experimental results indicate that TISC is superior to other joint clustering and alignment algorithms. In addition, simultaneous alignment and clustering is better than aligning the images first and then clustering the aligned images.

We also compare with DiSC, SSC, CASS, and LSR in terms of clustering accuracy on this database. The reason why we compare with these subspace clustering algorithms is that most of them have been evaluated in similar clustering tasks on the MNIST database. In order to have a fair comparison, robust alignment by sparse and low-rank decomposition (RASL) [48] is adopted to align all of the digits. RASL is chosen for two reasons. First, it is an unsupervised batch image alignment algorithm designed for unconstrained environment, which is state-of-the-art. Second, it also seeks an optimal set of transformations to recover the linear relationship among the misaligned data samples. Following [48], default parameters in RASL are used. The parameters of DiSC are set according to [4]. We run DiSC 20 times and report its mean clustering accuracy. As for SSC, LSR, and CASS, we try our best to tune the parameters. Table 2 shows the clustering accuracy of different subspace clustering algorithms with and without RASL.

From Table 2 we can see that the clustering performance of all the compared algorithms on the original digits is worse than that on the digits aligned by RASL. In addition, CASS and LSR achieve better clustering results than SSC and DiSC. This is mainly because CASS and LSR have denser within-cluster affinities than SSC and DiSC. However, the improvements of CASS and LSR before and after alignment are limited because they adopted simple combination of alignment

[2] Thanks Mattar et al. [19] for providing us with the digits they used.

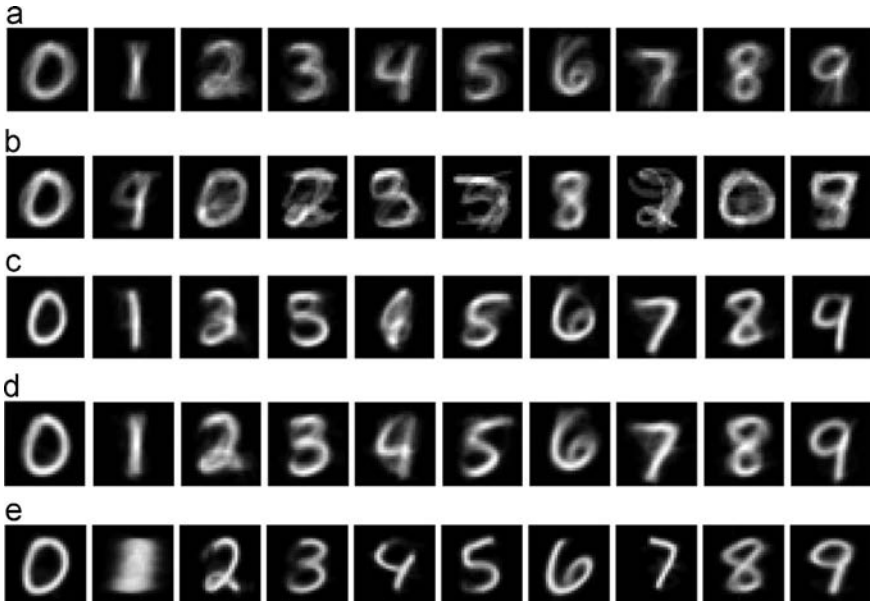

Fig. 3. The average digits obtained by different algorithms. (a) The average digits before alignment using the ground truth cluster labels. (b)–(e) The average digits after alignment using the estimated cluster labels by TIC, USAC, SSAC and TISC.

Table 2
Clustering accuracy of different subspace clustering algorithms with and without RASL on the MNIST database.

Algorithm	Accuracy (without RASL) (%)	Accuracy (with RASL) (%)
DiSC [4]	47.4	59.5
SSC [1]	60.5	68.5
CASS [10]	67.5	70.0
LSR [3]	68.5	71.5
TISC	**87.5**	

and subspace clustering. DiSC seems to have a relatively low clustering accuracy compared with other algorithms because it does not work well without sufficient data samples. When there are very few samples available on the subspaces, the performance of DiSC will deteriorate dramatically. Compared with SSC, CASS, and LSR, TISC achieves much better clustering results by benefiting from incorporating the transformation into subspace clustering.

4.4. Results on the CMU Multi-Pie database

We pursue an evaluation of our algorithm on the CMU Multi-PIE database [60], which contains a total number of 337 subjects with variations in pose, illumination, and expression. We choose 001-010 subjects from session 1 of this database. Each of the selected subjects has 20 frontal images taken under different illumination conditions with neural expression. The goal of this experiment is to show that our algorithm works stably and robustly to corruption and occlusion with practical variation in illumination. Both visual and quantitative results are reported in order to evaluate the performance of our algorithm. The visual result is verified by plotting the average face images before and after alignment. The clustering accuracy is evaluated by the Rand index with respect to the ground truth labels.

We synthetically perturb a random Euclidean transformation on each image. The angle of rotation is uniformly distributed in $[-10^{\circ}, 10^{\circ}]$ and the x and y translations are uniformly distributed in $[-3, 3]$ pixels. We also synthetically occlude the original input face image with a randomly chosen 40×40 patch, thus corrupting roughly 7 percent of all pixels in the face area. A total number of 60 among 200 images are randomly chosen with the synthetic corruption. Similarity transformation is used in our algorithm to align and cluster these misaligned images with variation of illumination, perturbation, and occlusion. The perturbed and occluded face images before and after alignment using our algorithm are shown in the supplementary material.

The average face images before and after the alignment are shown in Fig. 4. As shown in Fig. 4, the average face images after alignment using the estimated cluster labels are clearer and cleaner than the average faces before alignment using the ground truth labels. Different subspace clustering algorithms with and without RASL are used to compare with our algorithm. Table 3 shows the clustering accuracies of different algorithms. We can see that TISC achieves a much better clustering result than other popular subspace clustering algorithms despite the occlusion and noise. This experimental results further illustrate that our algorithm is robust to corruption, occlusion and illumination.

4.5. Results on the PubFig database

We also test our algorithm on the PubFig database [61], which is a real-world database consisting images of 200 people collected from the Internet. These images are taken in totally uncontrolled situations with large variation in pose, lighting, expression, scene, camera, imaging conditions, etc. We randomly choose 5 subjects with each of them containing 40 images. The face images are first detected by an OpenCV face detector [62] and then sent to different clustering algorithms. RASL algorithm is also chosen to align all of the face images. The clustering accuracies of different algorithms are reported in Table 4.

From Table 4 we can see that LSR and SSC perform slightly better than CASS and DiSC after alignment, which is consistent with previous experimental results. TISC achieves a better clustering result than other subspace clustering algorithms with or without RASL algorithm. This experimental results demonstrate the effectiveness of our algorithm on the real-world database. We will have a further analysis of our algorithm in the following part.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

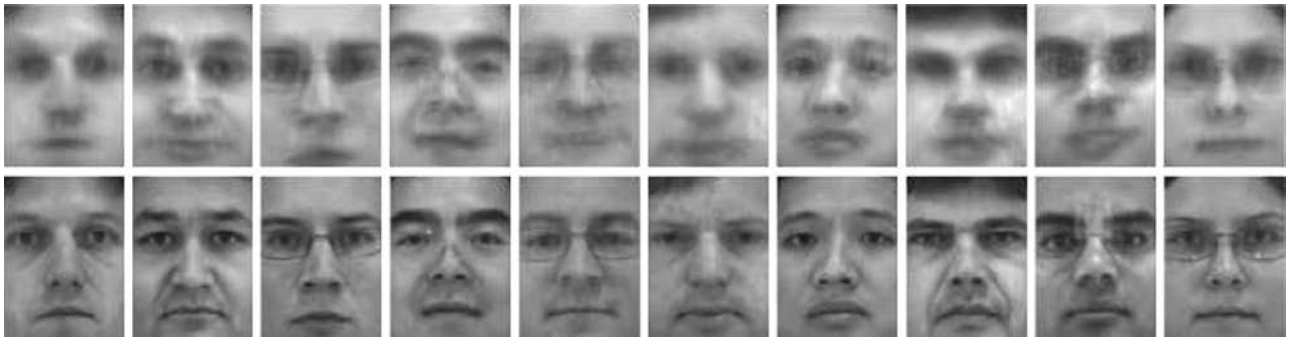

Fig. 4. Average face images before and after alignment with perturbation and occlusion on the CMU Multi-PIE database. *Top row*: The average face images before alignment using the ground truth cluster labels. *Bottom row*: The average face images after alignment using the estimated cluster labels.

Table 3
Clustering accuracies of different subspace clustering algorithms on the CMU Multi-PIE database.

Algorithm	Accuracy (without RASL) (%)	Accuracy (with RASL) (%)
DiSC [4]	30.5	43.5
SSC [1]	43.0	86.5
CASS [10]	57.5	84.5
LSR [3]	53.5	88.0
TISC	**93.0**	

Table 4
Clustering accuracies of different subspace clustering algorithms on the PubFig database.

Algorithm	Accuracy (without RASL) (%)	Accuracy (with RASL) (%)
DiSC [4]	53.3	64.5
SSC [1]	65.5	80.0
CASS [10]	62.0	76.5
LSR [3]	54.5	83.0
TISC	**85.5**	

4.6. Results on the LFW database

In this experiment, the LFW database is used to test the clustering and alignment performance of our algorithm. It is designed for studying the problem of unconstrained face recognition. It contains more than 13,000 face images which are collected from the Internet with large variations in pose, expression, and lighting. There are a total of 5749 subjects, but only 24 of them have more than 35 images. Most of the identities have only one image. 5, 10, 15, 20 subjects are randomly chosen from the LFW database with each of the subjects has 35 images. The experimental results include clustering accuracy and alignment performance. The clustering accuracy is evaluated by Rand-index with respect to a different number of subjects. There are no standard metrics to evaluate the alignment performance of this database. Note that face alignment itself is not an end goal in itself, but rather a step for producing better face recognition result. So we evaluate the alignment performance of TISC from two aspects in this experiment: direct visual impression about the alignment result and face recognition rate which is used as an indirect metric to evaluate the alignment performance.

The compared subspace clustering algorithms include DiSC, SSC, CASS, and LSR. All of the face images are first detected by an OpenCV face detector. Then the detected face images are used as the input of different subspace clustering algorithms. Considering that the above subspace clustering algorithms have not conducted similar experiments on the LFW database and most of them are not very robust to the nonlinear image-plane transformation, RASL is used to align all of the detected face images. The parameters of different algorithms are tuned carefully to achieve the best clustering results. SSC is difficult to run beyond 10 subjects in this experiment (similar to [4]). So only the clustering accuracies of 5 and 10 subjects are reported for comparison.

The clustering results of different algorithms on the original face images are shown in Fig. 5(a). We can see that SSC, CASS, and LSR almost get the same clustering accuracy. But none of them can successfully handle face images with large variations in pose, illumination, and expression. DiSC cannot deal with the face images in the LFW database due to the limited number of face images. The clustering accuracy of TISC is much better than its competitors. Even with a total number of 700 face images, its clustering accuracy is still 64%. Fig. 5(b) shows the clustering accuracies of different subspace clustering algorithms on the face images aligned by RASL. We can see that there are large improvements for CASS, SSC, and LSR on the well aligned LFW database. Such improvements are more notable than those on the MNIST database. This is because face images have a better subspace structure than handwritten digits. Face images from the same subject are similar to each other while handwritten digits from the same cluster may change greatly. As expected, TISC still achieves the highest clustering performance. This is due to the fact that alignment and clustering can rectify each other, which makes TISC more robust to various misalignment.

We present some of the well aligned face images in Fig. 6. The red bounding box indicates the face images detected by the OpenCV face detector and the green bounding box indicates face images aligned by TISC. From Fig. 6 we can see that TISC can automatically align the misaligned face images based on the underlying subspaces where they are drawn from. It seeks the optimal transformation parameters to encourage the linear relationship among the data samples. Even with the face images obtained by a coarse face detector, TISC can automatically cluster and align the face images, which is useful in real-world applications such as clustering face images in surveillance scenarios.

Face recognition rate is also used as a metric to evaluate the alignment performance of different algorithms. RASL and deep funneled algorithm are chosen to have a qualitative comparison with TISC with respect to face recognition rate. Deep funneled algorithm is chosen because it has been proved to produce better results for most face recognition algorithms over other unsupervised alignment algorithms, for example, funneled algorithm. The deep funneled images can be found on the online web set.[3] Two methods are used in this experiment to collect face images from the deep funneled images: the OpenCV face detector and the manually labeled bounding box. We manually label the outer eye

[3] http://www.vis-www.cs.umass.edu/lfw/

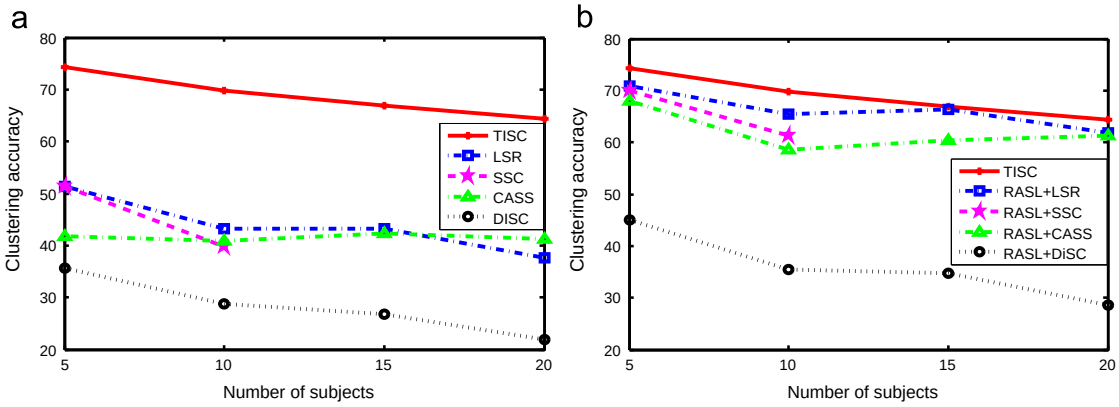

Fig. 5. Clustering accuracy of different subspace clustering algorithms with and without RASL on the LFW database. (a) Clustering accuracy on the original face images detected by the OpenCV face detector. (b) Clustering accuracy on the aligned face images (except for TISC, which is still using the original face images).

Fig. 6. Visual impression about the alignment performance of our algorithm (better viewed in color). The red bounding box indicates face images detected by the OpenCV Viola-Jones face detector. The green bounding box indicates face images aligned by our algorithm automatically. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

coordinates of the face images in the deep funneled images. Then a bounding box is used to capture the face images from the deep funneled images. With the face images from the original LFW database and the well aligned face images, five different kinds of face images are used as the input of the following face recognition algorithm.

The obtained face images are partitioned into a training set and testing set. Different number of face images per subject are selected for training and the remaining face images for testing. As for the recognition algorithm, Fisherface algorithm [63] is used in this experiment: all training face images are projected into the feature space of a 19 dimensional vector, and the nearest neighbor classifier was used to classify the face images in the testing set. We randomly choose 5, 10, 15, 20 face images per individual for training and repeat the experiment for 20 times. Then the face recognition result is exhibited in the form of the mean recognition rate. The mean recognition rate of different algorithms is shown in Fig. 7. As shown in Fig. 7, TISC achieves better face recognition rate

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

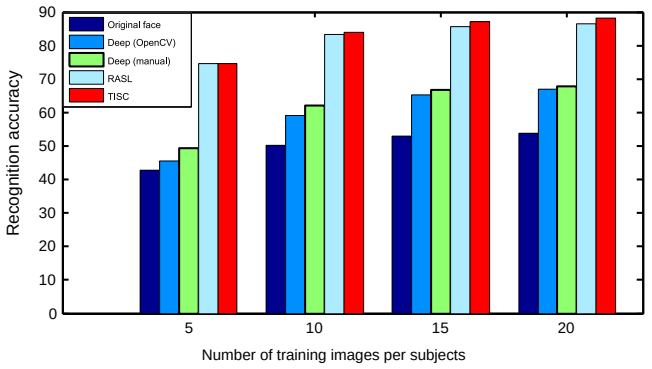

Fig. 7. Face recognition rate of different algorithms. TISC means face images aligned by our algorithm. RASL refers to face images automatically aligned by RASL. Deep (manual) represents face images manually labeled by a bounding box in the deep funneled LFW images. Deep (OpenCV) means face images from OpenCV face detector in the deep funneled LFW images. Original face represents face images from OpenCV face detector in the original LFW images.

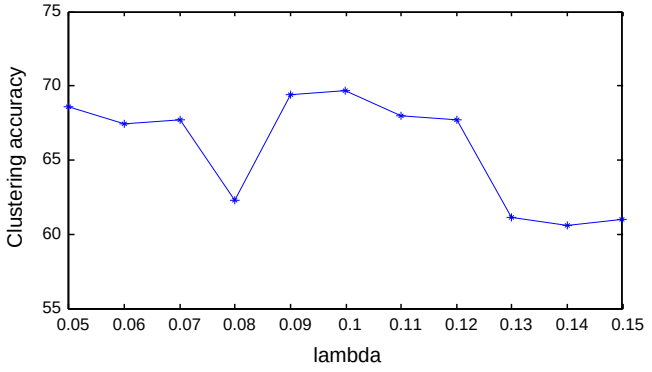

Fig. 8. Clustering accuracy as a function of parameter λ.

than the deep funneled face images whether using the face images from the OpenCV face detector or from the manually labeled bounding box. However, TISC performs only slightly better than RASL in terms of the recognition rate. The main reason is that both RASL and TISC seek an optimal set of image domain transformations to encourage linear correlation of all the images. Compared with RASL which is designed mainly for a single cluster, TISC performs better on multiple clusters. This experiment further validates that TISC can be used for clustering the unconstrained real-world database.

4.7. Parameter setting

In this section, we run the experiment on the LFW database (10 subjects from Section 4.6) to show how the regularization parameter λ affects clustering performance of TISC. The experimental setting is the same as to Section 4.6 except the parameter λ. We range the value of λ from 0.05 to 0.15. The clustering results are presented as a function of λ in Fig. 8. From Fig. 8, we observe that although there are variations of clustering accuracy when λ is set to different values, these variations are relatively small. There is a relatively large range for our algorithm to get higher clustering accuracy than others.

5. Conclusion

In this paper, we propose a transformation invariant subspace clustering framework. Specifically, a generalized cost function for simultaneously aligning and clustering data samples is presented. Then an efficient algorithm is proposed to recover the linear subspaces of misaligned data samples by incorporating transformation into the generalized framework. Experimental results show that the proposed TISC algorithm achieves much better clustering results than other joint alignment and clustering algorithms. This also demonstrates that alignment and subspace clustering are mutually dependent. In the future, we will work in speeding up TISC for large-scale clustering problems.

Conflict of interest

None declared.

Acknowledgements

The authors would like to thank the associate editor and the reviewers for their valuable comments and advice. We would also like to acknowledge the support by the National Basic Research Program of China (Grant Nos. 2012CB316300 and 2015CB352502), the National Natural Science Foundation of China (Grant No. 61273272 and Grant No. 61473289).

Appendix A

A.1. The solution for Eq. (16)

$$\|X^r - X^r Z\|_F^2 = \sum_{i=1}^{n} \left\| (I_i^r + J_i\Delta\tau_i) - \sum_{\substack{j=1 \\ j\neq i}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{ji} \right\|^2 = \left\| (I_s^r + J_s\Delta\tau_s) - \sum_{\substack{j=1 \\ j\neq s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{js} \right\|^2 + \sum_{\substack{i=1 \\ i\neq s}}^{n} \left\| (I_i^r + J_i\Delta\tau_i) - \sum_{\substack{j=1 \\ j\neq i,s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{ji} - (I_s^r + J_s\Delta\tau_s) Z_{si} \right\|^2. \quad (23)$$

Eq. (23) is simply a least squares problem. By taking partial derivative of the above Equation with respect to $\Delta\tau_s$, where we have utilized $Z_{ii} = 0$, we have:

$$\begin{aligned} & J_s^T \left[(I_s^r + J_s\Delta\tau_s) - \sum_{\substack{j=1 \\ j\neq s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{js} \right] - J_s^T \sum_{\substack{i=1 \\ i\neq s}}^{n} Z_{si} \left[(I_i^r + J_i\Delta\tau_i) - \sum_{\substack{j=1 \\ j\neq i,s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{ji} - (I_s^r + J_s\Delta\tau_s) Z_{si} \right] = J_s^T \left[(I_s^r + J_s\Delta\tau_s) - \sum_{\substack{j=1 \\ j\neq s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{js} \right] - J_s^T \sum_{\substack{i=1 \\ i\neq s}}^{n} Z_{si} (I_i^r + J_i\Delta\tau_i) \\ & + J_s^T \sum_{\substack{i=1 \\ i\neq s}}^{n} Z_{si} \sum_{\substack{j=1 \\ j\neq i,s}}^{n} (I_j^r + J_j\Delta\tau_j) Z_{ji} + J_s^T \sum_{\substack{i=1 \\ i\neq s}}^{n} Z_{si}^2 (I_s^r + J_s\Delta\tau_s) \\ & = J_s^T \left[\left(1 + \sum_{\substack{i=1 \\ i\neq s}}^{n} Z_{si}^2 \right) (I_s^r + J_s\Delta\tau_s) - \sum_{\substack{j=1 \\ j\neq s}}^{n} \left(Z_{js} + Z_{sj} - \left(\sum_{\substack{i=1 \\ i\neq j,s}}^{n} Z_{si} \right) Z_{ji} \right) (I_j^r + J_j\Delta\tau_j) \right] = 0. \end{aligned} \quad (24)$$

Fig. 9. The first 40 perturbed and occluded face images before and after alignment on the CMU Multi-PIE database. (a) Face images before alignment. (b) The corresponding face images after alignment.

So Eq. (24) becomes the following:

$$\left(1+\sum_{\substack{i=1\\ i\neq s}}^{n} Z_{si}^2\right)\left(J_s^T J_s\right)\Delta\tau_s - \sum_{\substack{j=1\\ j\neq s}}^{n}\left(Z_{js}+Z_{sj}-\left(\sum_{\substack{i=1\\ i\neq j,s}}^{n} Z_{si}\right)Z_{ji}\right)\left(J_s^T J_j\right)\Delta\tau_j$$

$$= -J_s^T\left[\left(1+\sum_{\substack{i=1\\ i\neq s}}^{n} Z_{si}^2\right)I_s^r - \sum_{\substack{j=1\\ j\neq s}}^{n}\left(Z_{js}+Z_{sj}-\left(\sum_{\substack{i=1\\ i\neq j,s}}^{n} Z_{si}\right)Z_{ji}\right)I_j^r\right], \quad s=1,2,\ldots,n. \tag{25}$$

Stacking Eq. (25) together, where $s=1,2,\ldots,n$, we can obtain the linear system for solving $\Delta\tau$.

A.2. More results on the CMU multi-pie database and the LFW database

Fig. 9 shows the first 40 perturbed and occluded face images before and after alignment using our algorithm. More visual results on the LFW database are shown in Fig. 10.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

Fig. 10. More visual impression about the alignment performance on the LFW database. The red bounding box represents face images detected by the OpenCV Viola-Jones face detector. The green bounding box represents face images aligned by our algorithm automatically. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

References

[1] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell 35 (11) (2013) 2765–2781.
[2] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell 35 (1) (2013) 171–184.
[3] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: European Conference on Computer Vision, 2012, pp. 347–360.
[4] V. Zografos, L. Ellis, R. Mester, Discriminative subspace clustering, in: Computer Vision and Pattern Recognition, 2013, pp. 2107–2114.
[5] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, Z. Lin, Correntropy induced l2 graph for robust subspace clustering, in: IEEE International Conference on Computer Vision, 2013, pp. 1801–1808.
[6] Y. Zhang, Z. Sun, R. He, T. Tan, Robust subspace clustering via half-quadratic minimization, in: International Conference on Computer Vision, 2013, pp. 3096–3103.
[7] R. He, Y. Zhang, Z. Sun, Q. Yin, Robust subspace clustering with complex noise, IEEE Trans. Image Process. 24 (11) (2015) 4001–4013.
[8] J. Feng, Z. Lin, H. Xu, S. Yan, Robust subspace segmentation with block-diagonal prior, in: Computer Vision and Pattern Recognition, 2014, pp. 3818–3825.
[9] C.-G. Li, R. Vidal, Structured sparse subspace clustering: a unified optimization framework, in: Computer Vision and Pattern Recognition, 2015.
[10] C. Lu, J. Feng, Z. Lin, S. Yan, Correlation adaptive subspace segmentation by trace lasso, in: IEEE International Conference on Computer Vision, 2013, pp. 1345–1352.
[11] H. Hu, Z. Lin, J. Feng, J. Zhou, Smooth representation clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3834–3841.
[12] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: International Conference on Machine Learning, 2010, pp. 663–670.
[13] C. Li, L. Lin, W. Zuo, S. Yan, J. Tang, Sold: sub-optimal low-rank decomposition for efficient video segmentation, in: Computer Vision and Pattern Recognition, 2015, pp. 5519–5527.
[14] D. Park, C. Caramanis, S. Sanghavi, Greedy subspace clustering, in: Advances in Neural Information Processing Systems, 2014, pp. 2753–2761.
[15] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[16] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, in: Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[17] B.J. Frey, N. Jojic, Transformation-invariant clustering using the em algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 1–17.
[18] X. Liu, Y. Tong, F.W. Wheeler, Simultaneous alignment and clustering for an image ensemble, in: International Conference on Computer Vision, 2009, pp. 1327–1334.
[19] M. Mattar, A. Hanson, E.G. Learned-Miller, Unsupervised joint alignment and clustering using Bayesian nonparametrics, in: Conference on Uncertainty in Artificial Intelligence, 2012, pp. 584–593.
[20] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, et al., Labeled faces in the wild: a database for studying face recognition in unconstrained environments, in: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, 2008.
[21] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[22] M. Cox, S. Sridharan, S. Lucey, J. Cohn, Least squares congealing for unsupervised alignment of images, in: Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[23] R. Martin, O. Arandjelović, Multiple-object tracking in cluttered and crowded public spaces, in: Advances in Visual Computing, 2010, pp. 89–98.
[24] Y. Ma, A.Y. Yang, H. Derksen, R. Fossum, Estimation of subspace arrangements with applications in modeling and segmenting mixed data, SIAM Rev. 50 (3) (2008) 413–458.
[25] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (gpca), IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1945–1959.
[26] Y. Liang, J.-H. Lai, P.C. Yuen, W.W. Zou, Z. Cai, Face hallucination with imprecise-alignment using iterative sparse representation, Pattern Recognit. 47 (10) (2014) 3327–3342.
[27] O. Arandjelović, R. Cipolla, Achieving robust face recognition from video by combining a weak photometric model and a learnt generic face invariant, Pattern Recognit. 46 (1) (2013) 9–23.
[28] J. Ho, M. Yang, J. Lim, K. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: Computer Vision and Pattern Recognition, vol. 1, 2003, pp. I–11.
[29] A. Gruber, Y. Weiss, Multibody factorization with uncertainty and missing data using the em algorithm, in: Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 1–707.
[30] Y. Sugaya, K. Kanatani, Geometric structure of degeneracy for multi-body motion segmentation, in: Statist. Methods Video Process., 2004, pp. 13–25.
[31] M. Tipping, C. Bishop, Mixtures of probabilistic principal component analyzers, Neural Comput. 11 (2) (1999) 443–482.
[32] U. Von Luxburg, A tutorial on spectral clustering, Statist. Comput. 17 (4) (2007) 395–416.
[33] J. Yan, M. Pollefeys, A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate, in: European Conference on Computer Vision, 2006, pp. 94–106.
[34] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Computer Vision and Pattern Recognition, 2009, pp. 2790–2797.
[35] X. Peng, Z. Yi, H. Tang, Robust subspace clustering via thresholding ridge regression, in: AAAI Conference on Artificial Intelligence, 2015.
[36] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[37] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, Y. Ma, Toward a practical face recognition system: robust alignment and illumination by sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2) (2012) 372–386.
[38] R. He, W.-S. Zheng, T. Tan, Z. Sun, Half-quadratic-based iterative minimization for robust sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 261–275.
[39] R. He, T. Tan, L. Wang, Robust recovery of corrupted low-rank matrix by implicit regularizers, IEEE Trans. Pattern Anal. Mach. Intell. 36 (4) (2014) 770–783.
[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[41] A. Vedaldi, G. Guidi, S. Soatto, Joint data alignment up to (lossy) transformations, in: Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[42] L. Lin, S.-C. Zhu, Y. Wang, Layered graph match with graph editing, in: Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[43] L. Lin, K. Zeng, X. Liu, S.-C. Zhu, Layered graph matching by composite cluster sampling with collaborative and competitive interactions, in: Computer Vision and Pattern Recognition, 2009, pp. 1351–1358.
[44] Q. Li, Z. Sun, R. He, T. Tan, Joint alignment and clustering via low-rank representation, in: IAPR Asian Conference on Pattern Recognition, 2013, pp. 591–595.
[45] L. Lin, X. Liu, S.-C. Zhu, Layered graph matching with composite cluster sampling, IEEE Trans. Pattern Anal. Mach. Intell. 32 (8) (2010) 1426–1442.
[46] Y. Xu, L. Lin, W.-S. Zheng, X. Liu, Human re-identification by matching compositional template with cluster sampling, in: IEEE International Conference on Computer Vision, 2013, pp. 3152–3159.
[47] Y. Peng, A. Ganesh, J. Wright, W. Xu, Y. Ma, Rasl: robust alignment by sparse and low-rank decomposition for linearly correlated images, in: Computer Vision and Pattern Recognition, 2010, pp. 763–770.
[48] Y. Peng, A. Ganesh, J. Wright, W. Xu, Y. Ma, Rasl: robust alignment by sparse and low-rank decomposition for linearly correlated images, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2233–2246.
[49] Z. Zhang, A. Ganesh, X. Liang, Y. Ma, Tilt: transform invariant low-rank textures, Int. J. Comput. Vis. 99 (1) (2012) 1–24.
[50] M. Yang, L. Zhang, D. Zhang, Efficient misalignment-robust representation for real-time face recognition, in: European Conference on Computer Vision, 2012, pp. 850–863.
[51] J. Huang, X. Huang, D. Metaxas, Simultaneous image transformation and sparse representation recovery, in: Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[52] Z. Huang, X. Zhao, S. Shan, R. Wang, X. Chen, Coupling alignments with recognition for still-to-video face recognition, in: International Conference on Computer Vision, 2013, pp. 3296–3303.
[53] W. Deng, J. Hu, J. Lu, J. Guo, Transform-invariant pca: a unified approach to fully automatic facealignment, representation, and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 36 (6) (2014) 1275–1284.
[54] Y. Wu, B. Shen, H. Ling, Online robust image alignment via iterative convex optimization, in: Computer Vision and Pattern Recognition, 2012, pp. 1808–1814.
[55] S. Baker, I. Matthews, Lucas-kanade 20 years on: a unifying framework, Int. J. Comput. Vis. 56 (3) (2004) 221–255.
[56] L. Lin, X. Wang, W. Yang, J. Lai, Learning contour-fragment-based shape model with and-or tree representation, in: Computer Vision and Pattern Recognition, 2012, pp. 135–142.
[57] E.G. Learned-Miller, Data driven image models through continuous joint alignment, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2) (2006) 236–250.
[58] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[59] B.D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, 1981.
[60] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, in: Automatic Face Gesture Recognition, 2008, pp. 1–8.
[61] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and Simile Classifiers for Face Verification, in: IEEE International Conference on Computer Vision, 2009.
[62] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[63] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006

Qi Li received the B.E. degree in automation from China University of Petroleum, Qingdao, China, in 2011. He is currently a Ph.D. candidate with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), China. His research interests focus on face preprocessing, computer vision and machine learning.

Zhenan Sun received the B.E. degree in industrial automation from Dalian University of Technology, Dalian, China, the M.S. degree in system engineering from Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in pattern recognition and intelligent systems from CASIA in 1999, 2002, and 2006, respectively. He is currently a Professor with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), China. His current research interests include biometrics, pattern recognition, and computer vision. He is a member of the IEEE and the IEEE Computer Society.

Zhouchen Lin received the Ph.D. degree in applied mathematics from Peking University in 2000. Currently, he is a professor at the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a chair professor at Northeast Normal University. Before March 2012, he was a leader researcher in the Visual Computing Group, Microsoft Research Asia. He was a guest professor at Shanghai Jiaotong University, Beijing Jiaotong University, and Southeast University. He was also a guest researcher at the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, machine learning, pattern recognition, and numerical computation and optimization. He serves as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Learning and International Journal of Computer Vision. He is a senior member of the IEEE.

Ran He received the B.E. and M.S. degrees in computer science from the Dalian University of Technology, in 2001 and 2004, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2009. Since 2010, he has been with the National Laboratory of Pattern Recognition, where he is currently a Full Professor. He currently serves as an Associate Editor of Neurocomputing (Elsevier), and serves on the Program Committee of several conferences. His research interests focus on information theoretic learning, pattern recognition, and computer vision.

Tieniu Tan received the B.Sc. degree in electronic engineering from Xi'an Jiaotong University, China, in 1984, and the M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, U.K., in 1986 and 1989, respectively. He is currently a Professor with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests include biometrics, image and video understanding, information hiding, and information forensics. He is a Fellow of IEEE and the IAPR (International Association of Pattern Recognition).

Please cite this article as: Q. Li, et al., Transformation invariant subspace clustering, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.02.006