VFSC: A Very Fast Sparse Clustering to Cluster Faces from Videos

Dinh-Luan Nguyen, Minh-Triet Tran

University of Science, VNU-HCMC, Ho Chi Minh city, Vietnam
1212223@student.hcmus.edu.vn
tmtriet@fit.hcmus.edu.vn

Abstract. Face clustering is a task to partition facial images into disjoint clusters. In this paper, we investigate a specific problem of face clustering in videos. Unlike traditional face clustering problem with a given collection of images from multiple sources, our task deals with set of face tracks with information about frame ID. Thus, we can exploit two kinds of prior knowledge about the temporal and spatial information from face tracks: sequence of faces in the same track and contemporary faces in the same frame. We utilize this forehand lore and characteristic of low rank representation to introduce a new light weight but effective method entitled Very Fast Sparse Clustering(VFSC). Since the superior speed of VFSC, the method can be adapted into large scale real-time applications. Experimental results with two public datasets (BF0502 and Notting-Hill), on which our proposed method significantly breaks the limits of not only speed but also accuracy clustering of state-of-the-art algorithms (up to 250 times faster and 10% higher in accuracy), reveal the imminent power of our approach.

1 Introduction

Object clustering is a work of organizing/grouping objects based on their characteristics. The characteristics here can be either low-level features, like raw input pixels, or high-level features extracted by a specific descriptor. Since clustering is one of the essential parts leading to success of other areas such as segmentation[1], tracking[2], recognition[3] and still challenged by noisy input data, it is really an allured field to be fully exploited.

Face clustering in video is a specific area of object clustering but it has special interests because it possesses wide range of applications in daily life, including auto tagging/naming character, video organization, video summarization, content retrieval, etc. Although several efforts [4,5] have been proposed to solve this task, it is still challenging and inspiring by the difficulties of real-life captured video input. For instance, the uncontrolled environments surrounding faces lead to the huge face appearances, erratic illumination, head position, pose and facial expression. Furthermore, occlusion and blur may occur during capturing faces by character's gestures or actions. Thus, these are the main reasons for face clustering not being regarded as a saturated area.

There is an existing misconception between face clustering and face recognition that we need to clarify before moving on. Face clustering, as mentioned above, is an unsupervised problem in which no labeled faces are given before processing. Whereas

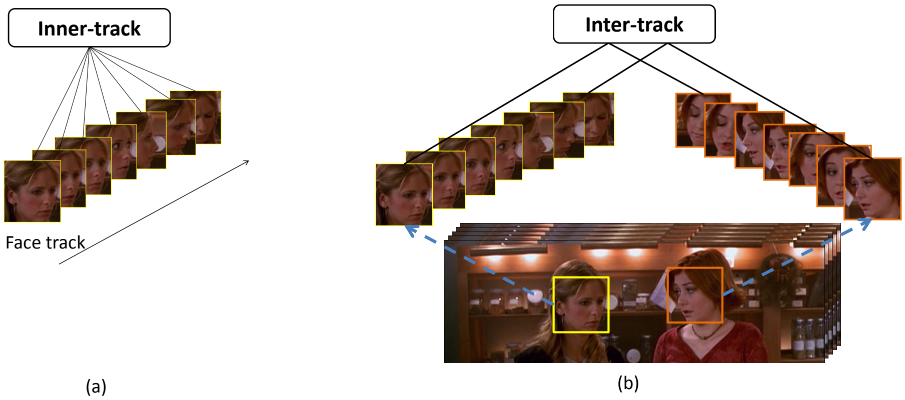

Fig. 1. Prior knowledge of face tracks in video extracted from "Buffy the Vampire Slayer" TV series. (a) Facial images in the same track must belongs to one person (Inner-track relation); (b) Two faces are concurrent in the same frame or two tracks overlap at least one frame are the possession of different people (Inter-track relation).

face recognition utilizes labeled images to train a classifier to return person face ID in the test set. This procedure is supervised problem.

In clustering faces retrieved from video, there is some information we can exploit. First, face images are not discrete. They usually come in sequences called face tracks, obtained by detection or tracking stage. In this paper, the input tracks for processing do not contain any contamination. In specific, all images from one track belong to one person and exclude any images from other. This leads to two other following characteristics [6]: *Inner-track*: images in one track belongs to the same person; and *Inter-track*: tracks having concurrent images are usually related to different people. These characteristics are useful to boost up the accuracy of the clustering process described in our proposed system. Figure 1 visualizes these two properties in clustering facial images in video scenario.

To deal with clustering problem, several works [7,8,9] have been proposed to split data into several clusters corresponding to relevant subspaces. The idea of representing one data point as the linear combination of others in the identical subspace [10,8] seems to get promising results with low-rank representation [8,9,11]. However, these approaches neglect information from face tracks and require high computation cost. Besides, the data-driven method with prior knowledge [6], called probabilistic constraint, successfully exploits the valuable characteristic of faces in video but still gets low accuracy in comparison with sparse methods.

Main contribution. Inspirited by these two main directions, in this paper, we present a new light weight approach inherited the idea of constraint and sparse representation methods to foster the accuracy and make our method applicable in real time. In specific, we create a sparse low-rank data representation integrated with prior knowledge from face tracks. This representation is a coefficient matrix which pulls images in one cluster closer and push images from different clusters far away. Sparsity appears for the linear performance of faces within tracks. In short, there are three main contributions in our work.

- Create a coalescence between sparse clustering and constraint knowledge characteristic.

- Explore an adaptive light weight method to significantly reduce computational complexity and boost up running time.

- Address the bias of evaluation protocol in previous works and proposed a new justified one.

Experiments in two public datasets show that our proposed method is superior than previous works and becomes a new state-of-the-art in this area in not only accuracy but also speed.

The rest of our paper is organized as follows: Section 2 reviews some related works on sparse subspace and constraint clustering methods. Our primary contributions for proposing a new effectively light weight clustering method for video faces in the wild are carefully discussed in section 3. Section 4 shows experimental results and comparisons to other state-of-the-art techniques on two face datasets from real-world videos. Finally, conclusion and our discussion are given in section 5.

2 Related Works

A lot of existing works [12,13,14] utilize data-driven methods, which focus on creating favorable distance metrics or transforming given data into new spaces, to foster inter-track differences. The first endeavor is clustering faces in videos by using affine invariant distance measurement [12]. Manifold-Manifold Distance (MMD) method proposed by Wang *et al.*[15] divides a nonlinear manifold into local linear subspaces. Specifically, from one subspace in the involved manifolds, it amalgamates distance between subspace pairs. Work of Arandjelovic and Cipolla [16] exploits the coherence of disparities between manifolds by grouping faces appearance in an anisotropic space. Another similar approach is to combine clustering with Bayesian method [17] to count dissimilar people in face images. In addition, Wolf *et al.*[18] propose Matched Background Similarity (MBGS) measurement to point out the distinction between face images having similar background. However, all mentioned works heavily depend on the data quality. Thus, they are unsteady in real-world videos. There are works [19,20] that make use of information from face tracks by modifying the distance matrix so that faces with inner-track relation come closer whereas faces in inter-track are thrust far away. Based on the Hidden Markov Random Fields (HMRF) model, HMRF-com [6], a probabilistic constrained clustering method, gets competitive results by exploiting prior knowledge of faces in video. However, this HMRF method and other related works [20,21] require high computation, which is inapplicable for real time systems.

Another kind of approach is to use subspace clustering methods. Several works [7,8,9] propose solutions for face clustering problem. While the problem in this paper is face images from videos in the wild, those works use the ideal or rectified images to process. Furthermore, they do not utilize the prior knowledge in video track effectively. Shijie *et al.*[22] propose Weighted Block-Sparse Low Rank Representation (WBSLRR) and claim that they exploit useful information from face images by integrating this information into their self-expressive matrix. This strategy seems to get promising results in comparison with other methods in spectral subspace clustering. However, all of these works require heavy parameter tuning for optimization. Thus, it takes huge time to process, depends on input data and is far from being applicable into real world. Based on these two trends of clustering mentioned above, it is essential to have a novel approach that does not rely on major parameter tuning but still produce good performance in real time. The following system proposed in Section 3 will resolve the inquiry.

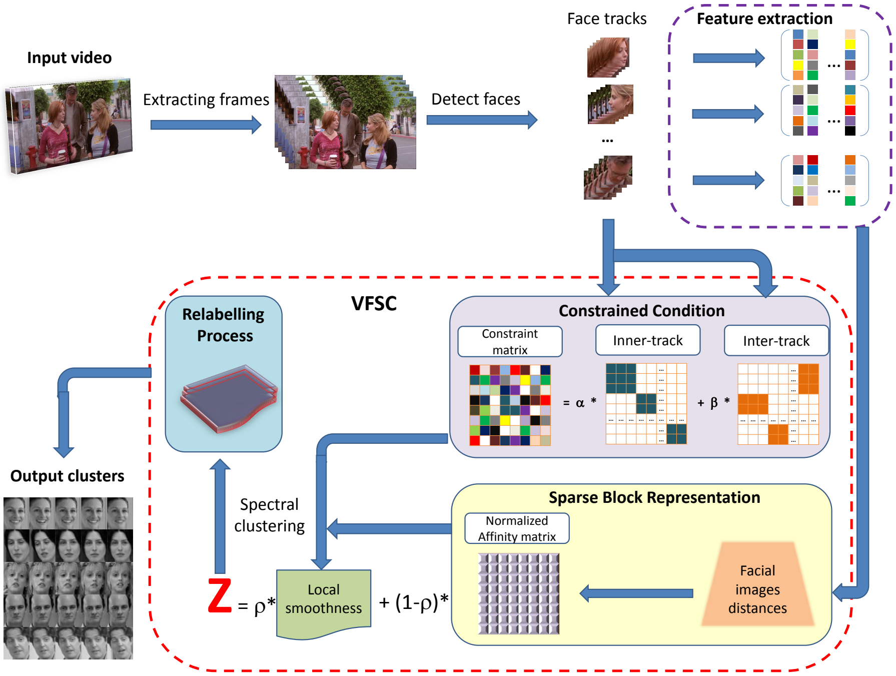

Fig. 2. An end-to-end system for face clustering in video. From given input video, we pre-process each frames to get face tracks and image features. Information from face tracks is an input for Constrained Condition module while image features are the input for Sparse Block Representation one. Constraint and affinity matrix returned from these module are used to create final data representation matrix Z. As a high accuracy face detector requires long processing time to complete its task, we do not apply it on every frame but only selected frames. Each detected face from interval frames is used as the input for the tracking process.

3 Proposed Method

3.1 Problem Specification

Given a set of unlabeled data

$$X = \{X_i\}_{i=1}^{m}$$

where $X_i \in R^{d \times n_i}$ denotes the i^{th} track in the total m face tracks, d is the dimension of each image vector. The goal is to divide X into K separate groups, where each group comprises tracks of the same subject. The number of faces in the dataset is $n = \sum_{i=1}^{m} n_i$, where n_i is the number of face images per track. The information of prior knowledge can be formulated into $n \times n$ symmetric matrices $W^{(in)} = \left\{W_{ij}^{(in)}\right\}$ and $W^{(it)} = \left\{W_{ij}^{(it)}\right\}; i, j = \overline{1..n}$, where $W_{ij}^{(in)} = 1$, $W_{ij}^{(it)} = -1$ if the i^{th} and j^{th} images belong to inner-track or inter-track relations, respectively.

3.2 System architecture

The modules of proposed method in an end-to-end clustering video face system is illustrated in Figure 2. Our system has 3 parts: sparse block representation, constrained condition, and relabelling process. To demonstrate the ability to integrate into end-to-end real-time application, from the given video, we apply face detector [23,24] to extract faces from interval frames. Since the high accuracy face detector take times to complete its task, we use these detected faces to stimulate for tracking process. An online tracking [25] is used to accelerate the extracting frame procedure.

Because the tracks'length returned by detection or tracking phase is different from others, we normalize it by subsampling $n_i = N; i = \overline{1..m}$ images from each track. For instance, given a L-length track, we choose N images at position $1 + k * (L/N); k = \overline{0..N-1}$ to represent this track. This way of sampling has the advantage of reserving the characteristics of the track in comparison with other methods such as choosing the first-N or last-N images in each track.

3.3 Sparse Block Representation

We inherit the idea of sparse subspace clustering [10], the problem can be regarded as finding the optimal $n \times n$ face relationship representation matrix Z such that $X = ZX$. Then, by applying spectral clustering method [11,7], we get the final face clustering result. As X is collected from tracks of face images, it can be seen as a union of linear subspaces [8]. Thus to solve the problem, we must solve the convex approximation of $rank(Z)$ [8,26]:

$$\min_{Z} \|Z\|_* \quad \text{s.t.} X = ZX$$

where $\|Z\|_*$ denotes the nuclear norm of Z. Traditional methods requires objective functions to adapt to the optimization of Z like $\|Z\|_1$ [10], $\|Z\|_F^2$ [7], or even adding new regularization terms of matrix itself [11,22]. However, in the paper, we propose to manage matrix Z based on input data X without heavy parameters for optimization. Thus, this technique has the advantage of being light weight and applicable in real time.

We apply PCA [27] to reduce the dimension of input data and lessen the computation cost. Based on the distances between data points, we construct a list of k-nearest neighbors for each point. Thus, we get n lists corresponding to n points in X. Specifically, let $\mathcal{N}_p(x_i)$ denotes the p nearest neighbors of x_i, matrix $D = \{d_{ij}\}; i, j = \overline{1..n}$, where

$$d_{ij} = \begin{cases} 0, & x_j \notin \mathcal{N}_p(x_i) \text{ or } i = j \\ exp(-d(x_i, x_j)/\sigma_i\sigma_j), & \text{otherwise} \end{cases} \tag{1}$$

is created to represent the affinity between data points. We construct $\sigma_i = d(x_i, x_p)$, where x_p is the p^{th} nearest neighbor of x_i and the distance is measured as L_2 norm. We use $L2$ for the default metric because it is robust with the range of input value X in general. Thus, we get the normalized affinity matrix $L = G^{-\frac{1}{2}}DG^{-\frac{1}{2}}$ where G is a diagonal matrix and $G_{ii} = \sum_{j=1}^{m} D_{ij}^2$. This matrix is integrated with constrain matrix in Section 3.4 to get final data representation matrix Z.

3.4 Constrained Condition

As in Section 3.1, by exploiting prior knowledge, we get $n \times n$ symmetric matrix $W^{(in)}$ and $W^{(it)}$ corresponding to inner-track and inter-track relation respectively. To improve the non-adaptive constrained condition in [6], which treats these matrices equally, we create new relation matrix W as follows:

$$W = \alpha W^{(in)} + \beta W^{(it)} \tag{2}$$

where α, $\beta \in \mathbb{R}$ are the adaptive weight of inner-track and inter-track relation respectively, which are dependent on the input dataset. The Equation 2 has the advantage that it does not treat $W^{(in)}$ and $W^{(it)}$ as the same importance. These weights are adaptive to the character of the input data. Since $W^{(in)}$ and $W^{(it)}$ are sparse, constrained matrix W also be sparse. Let $\mu = \arg\min_W W$ and $\nu = \arg\max_W W$, we get $\mu \leq W_{ij} \leq \nu$ where $w_{ij} \to \mu$ means the relationship between face image i^{th} and j^{th} is close to inter-track and $W_{ij} \to \nu$ means it in the vicinity of inner-track relation. Matrix W can be explained that it represents for the cost of prior knowledge.

Since $W^{(in)}$ and $W^{(it)}$ are sparse, matrix W is also sparse and heavily depends on the input dataset, it is not sufficient to be reliable to grant information for clustering scheme. Thus, we assuage this limitation by converting the constraint between pair wise images into soft one. Following the idea of the local smoothness method [28], we define the smooth constraint S as follows:

$$S = \left(1 - \frac{\gamma}{2}\right)^2 (I - \gamma L)^{-1} W (I - \gamma L)^{-1} \tag{3}$$

where $\gamma = \frac{1}{\exp\left(\frac{\alpha+\beta}{\exp(\alpha+\beta)}\right)}$ and I is an eye matrix. Equation 3 can be interpreted as the suppression of weight value α and β. Specifically, if α and β are too big or small, we get $\gamma \to 1$ so that S is a normalized smoothness. Otherwise, S is varied and depends on each weight value.

We convert the sparse low rank constraint, from prior knowledge, into the constrained matrix S. Since the magnitude of both sparse matrix L and constraint matrix

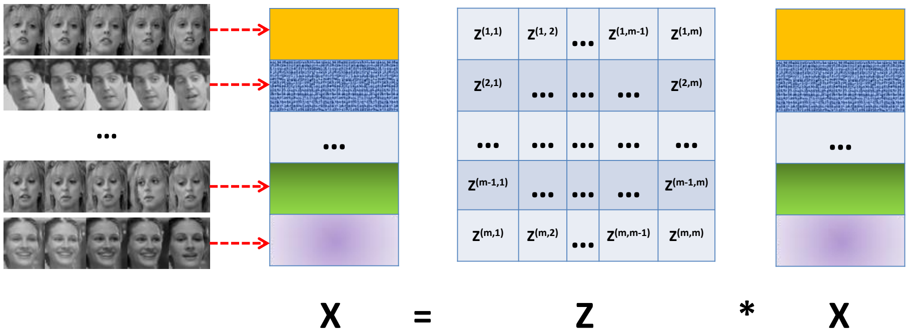

Fig. 3. Block sparse representation. This illustration depicts the relationship between input data X and self-data representation Z. Different color strips denote the different face tracks. Each block $Z^{(i,j)}$ denotes the coefficient of track i represented by track j.

S are too small, we get a natural combination of these matrices into final relationship matrix

$$Z = \exp\left(\frac{1}{2}(\rho S + (1-\rho)L)\right) \tag{4}$$

where $\rho \in (0,1)$ is the user-defined trade-off parameter.

3.5 Relabelling process

Since Z is the data representation matrix and includes the coefficients to exhibit the relationship between faces in track X^i and those in track X^j, it can be regarded as the matrix of sub-matrix $Z^{(i,j)} \in R^{n_i \times n_j}$: $Z = \left\{Z^{(i,j)}\right\}; i,j = \overline{1..m}$. The ideal of Z is that for the inner-track relation, $Z^{(i,j)}$ has the similar coefficient values if track i and j are from the same subject, otherwise, it is approximately zero; for the inter-track relation, the differences between coefficients of dissimilar clusters are must significant so that it can be easy for spectral clustering process. Figure 3 visualizes the ideal matrix Z as well as the clustering problem.

The spectral clustering [29,28] is widely used to obtain cluster labels from Z, so we inherit this approach as other works [11,9]. However, in those works, labels returning from spectral clustering process are taken by the most common label mode or even the raw ones. This has the disadvantage of neglecting the possibility of other labels if the distribution of those labels are roughly similar. Thus, we propose to consider the second and third mode of the label returned. We verify these modes because the correct labels do not always belong to the most frequent one. Specifically, if the proportion between the first mode and the others smaller than threshold value $\phi = 10\%$, we rerun our method with this corresponding mode for current track. Otherwise, we use the first mode. The details of our proposed system as well as relabelling process are summarized in Algorithm 1.

Algorithm 1 The proposed algorithm VFSC

```
Input: unlabeled face tracks $X = \{X_i\}_{i=1}^m$
Output: label for each track $T = \{T_1, T_2, \dots, T_m\}$
procedure VFSC ALGORITHM
    Compute distance matrix D as in 1
    Based on information from face tracks, calculate $W$ from $W^{(in)}$ and $W^{(it)}$ as in 2
    Compute $L = G^{-\frac{1}{2}} D G^{-\frac{1}{2}}$ where $G$ is a diagonal matrix and $G_{ii} = \sum_{j=1}^{m} D_{ij}^2$
    Smooth constraint matrix S is computed as in 3
    Get representation matrix $Z$ in 4
    Apply spectral clustering method to return raw labels $T' = \{T'_1, T'_2, \dots, T'_m\}$
    Get label distribution $M_i$ in each track $l'_i$
    for $T'_i \in T'$ do
        Get $M_i^{(1)}, M_i^{(2)}, M_i^{(3)}$ are percentages of the three most common labels in track $T'_i$
        if $M_i^{(1)} - M_i^{(r)} \leq \phi; r \in \{2, 3\}$ then
            Rerun VFSC with label of track $i$ is the $r$-th most common label in $T'_i$
        end if
        $T = Normalize(T')$
    end for
end procedure
```

4 Experiments

4.1 Datasets

We verify the merit of proposed system by using two public face datasets comprising Notting-Hill [6] and BF0502 [30]. The comparison here is not only the accuracy but also running time of all candidates. The Notting-Hill dataset is retrieved from "Notting Hill" movie, having 5 main characters comprising 76 tracks of 4660 faces. Whereas the BF0502 one has 17337 faces in 229 tracks belonging to 6 main casts in "Buffy the Vampire Slayer" TV series. Although it might be useful to extract high-level features by applying deep learning techniques [31,32], we still use raw RGB pixel as the input in system for fair comparison with other works.

To estimate the difficulty of each dataset, we create a distinction value based on face images in inner-track and inter-track relation. Specifically, we calculate the $L2$-norm distance between face images in the same track and those in dissimilar tracks. The distinction value is the ratio between image distance of inter-track over inner-track. This value can be interpreted that the smaller the value is, the more difficulty to cluster face tracks. Table 1 describes the information of both datasets. Not only has the dataset Notting-Hill smaller number of people as well as number of faces images and tracks but also larger distinction value in comparison with dataset BF0502, Notting-Hill is really easier for clustering. Therefore it gets high accuracy of clustering by state-of-the-art techniques and optimized clustering results by the proposed system described in Section 4.2 and 4.3.

Evaluation Protocol. In many existing works, the evaluation is still biased because they use the ratio between the number of correct clustered face images over the number of images (denoted as **"Acc-1"** in our experiments). Since the length of each track is

Table 1. Comparison between two face datasets in the real world. Symbol "#" denotes the phrase "the number of".

Dataset	#People	#Face	#Dimension	#Tracks	#Overlap	#Distinction value
BF0502	6	17737	1937	229	20	0.96
Notting-Hill	5	4660	18000	76	6	1.62

vary and mainly depends on the dataset. If one dataset has a long track while the number of track is small, this track will dominate the whole accuracy of this dataset.

Thus, we propose our metric that to compute the accuracy of clustering process that is the ratio of correct labeled track over the number of track (**"Acc-2"**). This evaluation will get rid of the unbalance between tracks' lengths. However, we also compare the accuracy between our proposed method with all previous works on both evaluation metrics.

Besides, since the output of methods having optimization process is varied, we repeat each works 30 times and report the mean accuracy and its standard deviation. To demonstrate the ability being integrated into real-life application, running time is also compared. All methods are executed on the same environment for fair comparison.

Baseline clustering methods. We compare our proposed method with state-of-the-arts in various kind of approaches in clustering problem to verify the superiority of our VFSC system. Specifically, the following are the types of algorithms that we consider in the comparison with our proposed method in experiments:

- *Traditional clustering:* We utilize K-means [21] as a baseline with two approaches given in [6]. No useful information of inner and inter-track is exploited in these methods. Firstly, using K-means to cluster processed dataset by PCA is called "K-means 1". Secondly, "K-means 2" indicates the using K-means in Stage 2 of Algorithm 2 in [6].

- *Constrained clustering:* Gaussian mixture models combines with image constraints is the idea of the Penalized Probabilistic Clustering (PPC) [19] method we use in comparison. Besides, following the advice of [6], we set the mixture coefficient parameter π in [19] to $\frac{1}{m}$ and size of all constraints in PPC is equal to 1.

- *Metric learning based:* In this type of approach, since the work of [33] is too slow to apply in the real-life application, we compare with the unsupervised logistic discriminant metric learning (ULDML) method [20]. There are also two settings called "ULDML-cl", a complete link hierarchical clustering method based on distance matrix between face tracks, and "ULDML-km", a K-means approach to utilize learned metric.

- *Hidden Markov Random Fields based:* A probabilistic constrained clustering approach called HMRF-com [6] is taken into consideration and gives out promising result from prior knowledge.

- *Compressed sensing based subspace clustering:* These methods have the same goal that optimize sparse matrix Z such as Sparse Subspace Clustering (SSC) [10], Least Squares Regression (LSR) [7], Correlation Adaptive Subspace Segmentation (CASS) [9], Low Rank Representation (LRR) [8], Low Rank Sparse Subspace Clustering (LRSSC) [11] and Weighted Block-Sparse Low Rank Representation (WBSLRR) [22]. However, the differences between those methods are the approaches to regularize on matrix Z in their objective function.

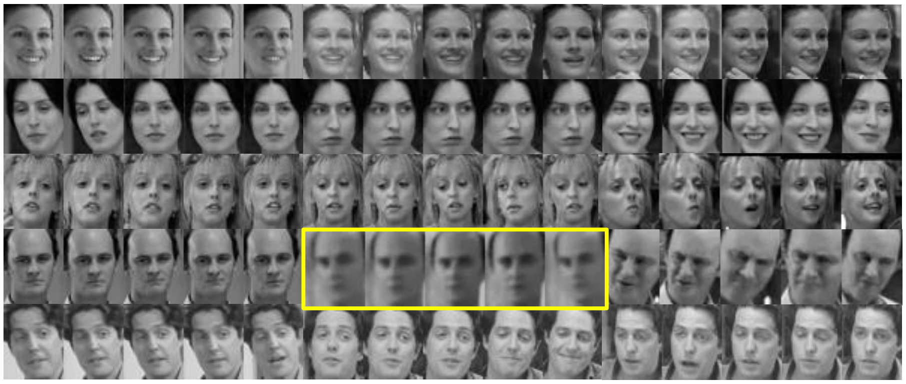

Fig. 4. Difficult track in Notting-Hill dataset. This track is blur and captured in low condition so that it is the big challenge to completely cluster Notting-Hill dataset.

4.2 Comparison with non sparse method

The results of the comparison between state-of-the-art techniques in non sparse subspace clustering and our proposed method are shown in Table 2. There are three configurations in our comparison: using inter-track relation only, inner-track only and both. From the table, we can observe that the proposed method surpasses all existing techniques in both datasets in all configurations. Specifically, in comparison with the second best method HMRF-com, we significantly increase the clustering accuracy: about 17.43% increases in inner-track (from 47.77% to 65.20%), 17.54% (from 48.83% to 66.37%) increases in inter-track and 16.67% (from 50.30% to 67.07%) increases in the whole setting on BF0502 dataset. ULDML-cl method reveals the similar characteristic like HMRF-com, which is the combination of both inner and inter-track help boost up the accuracy overall. Results of K-means 2 are better than K-means 1 in both datasets. However, they are still far from the satisfaction in comparison of accuracy.

In Notting-Hill dataset, the overall accuracy of all techniques seem to be high because it is regarded as an "easy" one based on the comparison of two datasets in Table 1 and analysis in Section 4.1. That is the reason why the proposed method approaches the optimal value (98.68%) clustering on this dataset while only got 66.37% on BF0502 dataset. However, there is still one track in Notting-Hill dataset is wrongly labeled. As showed in Figure 4, there are wrongly clustered samples in the forth row. Those are blurrier and noisier than others. This hazy characteristic leads to wrong clustered in spectral clustering process. Thus, it is very difficult to get 100% right label unless tuning parameters. Besides, the accuracy evaluation metric Acc-2 is usually lower than Acc-1 in both datasets. This phenomenon can be explained that easily clustered facial images are belonged to long track. Specifically, in 5 clusters of Notting-Hill dataset, HMRF-com can achieve up to 90.66%, 70.72%, 100%, 93.95% and 43.47% respectively in accuracy. However, since the number of images of the second and the fifth cluster has only nearly 30% of the whole dataset, the overall accuracy of HMRF-com is approximately 84.39%. In Acc-2 evaluation, long tracks and short tracks are equaly

Table 2. Comparison with non sparse clustering methods

Setting			**Methods**						
			Kmeans		PPC	ULDML		HMRF	VFSC
			1	2	[19]	km	cl	com [6]	(Proposed)
BF0502	**Acc-1**	Inner	39.31 ± 4.51	42.05 ± 5.45	43.64 ± 4.61	29.05 ± 2.84	39.01 ± 0.00	47.77 ± 3.31	**65.20 ± 0.00**
		Inter	39.31 ± 4.51	42.05 ± 5.45	38.22 ± 3.02	39.61 ± 1.42	47.97 ± 0.00	48.83 ± 4.05	**66.37 ± 0.00**
		Both	39.31 ± 4.51	42.05 ± 5.45	42.54 ± 3.98	41.62 ± 0.00	49.29 ± 0.00	50.30 ± 2.73	**67.06 ± 0.00**
	Acc-2	Inner	37.12 ± 2.62	39.30 ± 3.49	39.77 ± 2.94	31.75 ± 3.39	38.27 ± 0.00	45.62 ± 2.01	**58.43 ± 0.00**
		Inter	37.12 ± 2.62	39.30 ± 3.49	37.12 ± 3.57	37.82 ± 0.00	47.01 ± 0.00	46.61 ± 1.94	**62.44± 0.00**
		Both	37.12 ± 2.62	39.30 ± 3.49	40.11 ± 4.03	39.94 ± 0.79	48.28 ± 0.00	48.52 ± 3.49	**66.37± 0.00**
	Time(s)		72.06	87.93	61.46	437.32	410.56	119.26	**2.37**
Notting-Hill	**Acc-1**	Inner	69.16 ± 3.22	73.43 ± 8.12	79.71 ± 2.14	72.66 ± 12.78	51.72 ± 0.00	81.33 ± 0.43	**97.83 ± 0.00**
		Inter	69.16 ± 3.22	73.43 ± 8.12	77.05 ± 3.12	73.87 ± 5.98	35.91 ± 0.00	82.36 ± 2.67	**97.59 ± 0.00**
		Both	69.16 ± 3.22	73.43 ± 8.12	78.88 ± 5.15	73.18 ± 8.66	36.87 ± 0.00	84.39 ± 1.47	**98.73 ± 0.00**
	Acc-2	Inner	69.73 ± 5.26	71.05 ± 2.63	77.63 ± 1.31	70.13 ± 3.57	48.68 ± 0.00	79.26 ± 3.59	**97.36± 0.00**
		Inter	69.73 ± 5.26	71.05 ± 2.63	71.85 ± 6.57	70.94 ± 3.48	32.89 ± 0.00	80.01 ± 2.83	**97.36± 0.00**
		Both	69.73 ± 5.26	71.05 ± 2.63	72.37 ± 7.89	71.03 ± 2.11	42.10 ± 0.00	81.94 ± 1.16	**98.68± 0.00**
	Time(s)		14.77	15.08	10.03	84.24	89.57	19.58	**0.83**

in computing accuracy so that it is unbiased by input datasets. Although all previous works significantly degrade the clustering results while applying Acc-2 evaluation, our proposed method still reserve the high accuracy in both datasets.

Furthermore, the proposed method is superior in not only clustering accuracy but also speed. Our performance on BF0502 dataset only takes 2.37 seconds to finish clustering while the others take up to 119.26 (HMRF-com) or even 437.32 (ULDML-km) seconds to complete this task. Since we are based on sparse representation and computation and do not need any optimization parts, the running time is fast enough to be integrated in the real-life application. Further discussion on the running time is presented in Section 4.3 when compared with other state-of-the-art in sparse clustering methods. In general, the experiment results clearly reveal that VFSC has better usage of prior knowledge (i.e. the inner-track and inter-track relation) of facial images in video than other existing works.

4.3 Comparison with sparse low-rank method

The aim of this comparison is to prove superior robustness and speed of proposed method. Table 3 reports the comparison of mean accuracy in two evaluation metrics as well as standard deviations and running times of six subspace clustering works and the proposed method, where $\|Z\|_F = \left(\sum_{i,j} Z_{i,j}^2\right)^{\frac{1}{2}}$ denotes the Frobenius norm of Z. In general, subspace clustering is better than the non sparse clustering methods given in Section 4.2 in accuracy but it is too slow because of heavy computation. We can clearly notice that the the proposed VFSC system significantly outperform all subspace clustering state-of-the-arts. Like the explanation in Section 4.2, accuracy achieved in Acc-2 evaluation metric is lower than one in Acc-1 for all works except the proposed VFSC. Specifically, VFSC gets up to 4.30% and 2.44% in Acc-1 and 11.35% and 5.26% in Acc-2 higher than the WBSLRR, which is regarded as the best method for face clustering, in comparison on BF0502 and Notting-Hill dataset respectively. Since WBSLRR has $\Omega(Z)$ for their regularization part to encourage the block-sparsity of Z, WBSLRR's

Table 3. Comparison with sparse clustering method

Methods	**Regularization on Z**	**BF0502**			**Notting-Hill**		
		Acc-1	**Acc-2**	**Time**	**Acc-1**	**Acc-2**	**Time**
LSR [7]	$\|Z\|_F^2$	50.19 ± 1.93	46.72 ± 2.62	129.61	89.89 ± 0.00	85.52 ± 0.00	8.27
SSC [10]	$\|Z\|_1$	36.52 ± 0.91	33.18 ± 1.74	24558.27	75.50 ± 7.90	65.78 ± 6.57	2892.00
LRR [8]	$\|Z\|_*$	51.17 ± 2.94	48.03 ± 3.05	1201.08	93.11 ± 0.00	88.15 ± 0.00	34.07
CASS [9]	$\sum_{i=1}^n \|Xdiag(Ze_i)\|$	N/A	N/A	N/A	93.18 ± 0.00	90.78 ± 0.00	29672.40
LRSSC [11]	$\|Z\|_* + \gamma\|Z\|_1$	58.08 ± 5.37	54.14 ± 1.31	8218.24	94.03 ± 0.00	92.13 ± 0.00	538.51
WBSLRR [22]	$\|Z\|_* + \gamma\Omega(Z)$	62.76 ± 1.10	55.02 ± 3.49	692.66	96.29 ± 0.00	93.42 ± 0.00	201.93
VFSC (Proposed)	$\exp\left(\frac{1}{2}(\rho S + (1-\rho)L)\right)$	**67.06± 0.00**	**66.37± 0.00**	**2.37**	**98.73± 0.00**	**98.68±0.00**	**0.83**

result is better than LRR. Besides, the general sparsity neglecting the prior information between face tracks leads LRSSC get only 58.08% in BF0502 and 94.03% in Notting-Hill dataset. LSR and SSC method are just the special cases of LRR and LRSSC so that they has poor performance, however, they are still better than some non sparse methods described in Section 4.2.

Besides the accuracy in clustering, system's robustness is also important. This characteristic can be reflected from the standard deviation. For instance, a small standard deviation reveals good robustness whereas a big one means the corresponding method is easily fluctuated and fragile, depending on the input dataset. As can be observed from Table 2 and Table 3, sparse subspace clustering has the smaller standard deviation than the non sparse ones in Notting-Hill dataset. However, they are nearly equal in BF0502 dataset because this dataset has more faces and face tracks with small distinction value, which is not easy to solve completely. On the contrary to all of existing works, the proposed VFSC system presents zero standard deviation in both datasets with different configurations. One explanation is that we do not have regularization terms in constructing data representation matrix Z. Although LRSSC method gets zero standard deviation in Notting-Hill dataset, it get up to 5.37% in variation in BF0502 dataset. It can be explained that BF0502 is more difficult than the other. However, SSC method shows the contradiction. It achieves only 1.74% (Acc-2) standard deviation in BF0502 while jumping up to 6.57% in Notting-Hill dataset. Hence, SSC is easily fluctuate with the given data. Thus, our method, whose standard deviations are zero in all setting, is more robust and does not depend on the difficulty of input dataset.

Furthermore, our proposed VFSC method also shows the superior in speed in comparison with state-of-the-art of sparse techniques. In specific, we boost up running time from 129.61/8.27 seconds (LSR) to 2.37/0.83 seconds in BF0502 and Notting-Hill dataset respectively. Although LSR is the fastest among existing works, its accuracy is one of the worsts. Furthermore, while collating with WBSLRR on these two datasets, our proposed method still surpass this method about approximately 10% and 5% in accuracy (Acc-2) and 250 times faster in speed.

When comparing sparse and non sparse clustering method, we address the following comment: Non sparse methods prioritize the speed rather than clustering accuracy while sparse subspace clustering ones focus on the accuracy by adjusting parameters to satisfy regularization constraints but neglect the running time. The proposed VFSC system joins the advantages and assuage the defects of both trends. Therefore VFSC achieves superior results in accuracy and speed of clustering faces in video problems.

5 Conclusion

This paper presents VFSC system, a light weight novel approach to handle face clustering in video by utilizing the prior useful knowledge from face track and characteristic of sparse data representation. An end-to-end system for adapting into real-life application is also discussed. New evaluation metric is also given to solve the bias of existing works which depends largely on the input dataset. The experimental results reveals the truth that our method outperforms all state-of-the-art techniques in not only sparse but also non sparse clustering trends and become a new state-of-the-art method in clustering area. Since the swiftness, stability and high accuracy of our method while tackling face clustering problem, it is applicable to be adapted into very large scale tasks. Finally, for the future work, the accuracy of our system can be further improved by applying deep learning techniques in the post-processing stage.

References

1. Yao, H., Duan, Q., Li, D., Wang, J.: An improved k-means clustering algorithm for fish image segmentation. Mathematical and Computer Modelling **58** (2013) 790–798
2. Kang, Z., Landry, S.J.: An eye movement analysis algorithm for a multielement target tracking task: Maximum transition-based agglomerative hierarchical clustering. Human-Machine Systems, IEEE Transactions on **45** (2015) 13–24
3. Huang, Z., Wang, R., Shan, S., Chen, X.: Projection metric learning on grassmann manifold with application to video based face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 140–149
4. Aggarwal, C.C., Reddy, C.K.: Data clustering: algorithms and applications. CRC Press (2013)
5. Sang, J., Xu, C.: Robust face-name graph matching for movie character identification. Multimedia, IEEE Transactions on **14** (2012) 586–596
6. Wu, B., Zhang, Y., Hu, B.G., Ji, Q.: Constrained clustering and its application to face clustering in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3507–3514
7. Lu, C.Y., Min, H., Zhao, Z.Q., Zhu, L., Huang, D.S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: Computer Vision–ECCV 2012. Springer (2012) 347–360
8. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 171–184
9. Lu, C., Feng, J., Lin, Z., Yan, S.: Correlation adaptive subspace segmentation by trace lasso. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1345–1352
10. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 2790–2797
11. Wang, Y.X., Xu, H., Leng, C.: Provable subspace clustering: When lrr meets ssc. In: Advances in Neural Information Processing Systems. (2013) 64–72
12. Fitzgibbon, A., Zisserman, A.: On affine invariant clustering and automatic cast listing in movies. In: Computer VisionECCV 2002. Springer (2002) 304–320
13. Fitzgibbon, A.W., Zisserman, A.: Joint manifold distance: a new approach to appearance based clustering. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. Volume 1., IEEE (2003) I–26

14. Hu, Y., Mian, A.S., Owens, R.: Sparse approximated nearest points for image set classification. In: Computer vision and pattern recognition (CVPR), 2011 IEEE conference on, IEEE (2011) 121–128
15. Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-manifold distance with application to face recognition based on image set. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
16. Arandjelović, O., Cipolla, R.: Automatic cast listing in feature-length films with anisotropic manifold space. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 2., IEEE (2006) 1513–1520
17. Prince, S.J., Elder, J.H.: Bayesian identity clustering. In: Computer and Robot Vision (CRV), 2010 Canadian Conference on, IEEE (2010) 32–39
18. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 529–534
19. Lu, Z.L., Leen, T.K.: Penalized probabilistic clustering. Neural Computation **19** (2007) 1528–1567
20. Cinbis, R.G., Verbeek, J., Schmid, C.: Unsupervised metric learning for face identification in tv video. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1559–1566
21. Bishop, C.M.: Pattern recognition. Machine Learning (2006)
22. Xiao, S., Tan, M., Xu, D.: Weighted block-sparse low rank representation for face clustering in videos. In: Computer Vision–ECCV 2014. Springer (2014) 123–138
23. Nguyen, D.L., Nguyen, V.T., Tran, M.T., Yoshitaka, A.: Adaptive wildnet face network for detecting face in the wild. In: Eighth International Conference on Machine Vision, International Society for Optics and Photonics (2015) 98750S–98750S
24. Nguyen, D.L., Nguyen, V.T., Tran, M.T., Yoshitaka, A.: Boosting speed and accuracy in deformable part models for face image in the wild. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), IEEE (2015) 134–141
25. Zhang, K., Zhang, L., Yang, M.H.: Fast compressive tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014) 2002–2015
26. Zeng, Z., Chan, T.H., Jia, K., Xu, D.: Finding correspondence from multiple images via sparse and low-rank decomposition. In: Computer Vision–ECCV 2012. Springer (2012) 325–339
27. Jolliffe, I.: Principal component analysis. Wiley Online Library (2002)
28. Lu, Z., Ip, H.H.: Constrained spectral clustering via exhaustive and efficient constraint propagation. In: Computer Vision–ECCV 2010. Springer (2010) 1–14
29. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in neural information processing systems. (2004) 1601–1608
30. Everingham, M., Sivic, J., Zisserman, A.: Hello! my name is... buffy"–automatic naming of characters in tv video. In: BMVC. Volume 2. (2006) 6
31. Nguyen, D.L., Nguyen, V.T., Tran, M.T., Yoshitaka, A.: Deep convolutional neural network in deformable part models for face detection. In: Image and Video Technology. Springer (2015) 669–681
32. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 437–446
33. Vretos, N., Solachidis, V., Pitas, I.: A mutual information based face clustering algorithm for movie content analysis. Image and Vision Computing **29** (2011) 693–705