Finding Correspondence from Multiple Images via Sparse and Low-rank Decomposition

Zinan Zeng[1,2], Tsung-Han Chan[2], Kui Jia[2], and Dong Xu[1]

[1]School of Computer Engineering, Nanyang Technological University, Singapore
[2]Advanced Digital Sciences Center, Singapore
{znzeng,dongxu}@ntu.edu.sg, {Th.chan,Chris.jia}@adsc.com.sg

Abstract. We investigate the problem of finding the correspondence from multiple images, which is a challenging combinatorial problem. In this work, we propose a robust solution by exploiting the priors that the rank of the ordered patterns from a set of linearly correlated images should be lower than that of the disordered patterns, and the errors among the reordered patterns are sparse. This problem is equivalent to find a set of optimal partial permutation matrices for the disordered patterns such that the rearranged patterns can be factorized as a sum of a low rank matrix and a sparse error matrix. A scalable algorithm is proposed to approximate the solution by solving two sub-problems sequentially: minimization of the sum of nuclear norm and l^1 norm for solving relaxed partial permutation matrices, followed by a binary integer programming to project each relaxed partial permutation matrix to the feasible solution. We verify the efficacy and robustness of the proposed method with extensive experiments with both images and videos.

Key words: Feature correspondence, partial permutation, low rank and sparse matrix decomposition.

1 Introduction

Finding visual pattern correspondence across images or video sequences is a long-standing problem in computer vision. It facilitates a wide range of vision applications such as object recognition [1], 3D reconstruction [2, 3] and image matching [4, 5]. To date, most existing works focus on finding the feature correspondence between two images. The works in [1, 6–9] formulated it as graph matching problem, in which features were modeled as graph nodes while geometric relations between features were modeled as graph edges and efficient algorithms based on spectral technique were widely used to solve the problem. Linear programming model was also proposed to solve the correspondence problem [10–12], in which the geometric invariances were expressed as affine functions and constraints and they were incorporated into a linear model. In addition, Cho *et al.* [13] proposed to construct clusters of mutually coherent feature correspondence while eliminating outlier from candidate correspondence via hierarchical agglomerative clustering. Barnes *et al.* [14] proposed a algorithm based on kd-tree to efficiently find the dense matches.

It is also crucial to find the feature correspondence across video sequence. For example, in video action recognition, the trajectory of the key points can be easily found if features correspondence is established. In video segmentation, it relies on the well established correspondence among features to effectively divide the video into smaller clips. It is possible to directly apply the above mentioned methods to find correspondence among multiple images, however, this generally leads to suboptimal solutions because information in other images is not used for regularizing this pair of correspondence. In addition, propagating the result for the next pair of correspondence is error prone. On the other hand, very few researches focus on simultaneously finding correspondence among multiple images. In particular, the rank constraint was enforced in [15] to recursively compute the correspondence for one image at a time, but this greedy algorithm suffered from error propagation and it was not robust against noise and outliers; A graph matching approach was proposed in [16] to learn an embedding representation for all images and the k-means algorithm was then applied to find the correspondence, but it was difficult to enforce the bijective property of the correspondence and it did not have any mechanism to identify outliers.

Finding the global correspondence for features/patches (referred to as patterns) from multiple images can be formulated as a combinatorial NP-hard problem, in which a set of partial permutation matrices need to be found. Meanwhile, if each pattern is exactly the same across images, the matrix formed by the well corresponded patterns will be *low rank*, ideally rank one. However, if there exist variations such as occlusion, rotation and non-rigid transformations in images, the matrix of the well corresponded patterns might have an unknown rank higher than one. Therefore it is more appropriate to search for the permutation matrices that minimize the rank of a matrix formed by the well corresponded patterns. The pioneering works in [17, 18, 15] exploited this property and a greedy algorithm was proposed to recursively find one partial permutation at a time for correspondence. These works motivate us to employ the rank constraint as a criterion to effectively search optimal partial permutation matrices for the correspondence problem among multiple images.

Some recent developments for low rank representation have also incorporated the sparse error term into their frameworks to cope with corruption and occlusion that inevitably exist in images and videos [19–21]. Motivated by these works, we propose a robust and scalable method based on low-rank and sparse representation to find the optimal partial permutation matrices. Compared with the existing methods for feature correspondence, our proposed method has three major contributions: i) The proposed method formulates the challenging combinatorial correspondence task as a low-rank and sparse matrix decomposition that can be efficiently solved by our proposed algorithm. ii) The proposed method operates in a batch mode, in which all the images are simultaneously taken into consideration to determine the global optimal correspondence. Even if a misalignment happens, it would not be propagated to the subsequent frames. iii) The proposed method has a robust built-in mechanism to effectively cope with corruption and occlusion, and identify outliers.

2 Related Works

Finding feature correspondence between two images is a well-studied research topic. Different formulations that explore feature similarity and spatial constraint have been proposed. Meanwhile, these methods are either difficult or ineffective when used for finding correspondence for multiple images. To solve this problem, Marwan and Ahmed [16] proposed a graph matching approach to learn an Euclidean embedding representation via minimizing an objective function that promoted coherence between features similarity with their spatial arrangement, where the objective function was reduced to Laplacian embedding and was solved efficiently by only one eigen decomposition. The final correspondence could be obtained by using k-means algorithm. Although it is efficient, as argued in [7], such spectral technique is sensitive to noise and outliers.

The proposed method is motivated from the work by Ricardo *et al.* [15]. They proposed to enforce the rank constraint to find correspondence for multiple images. Specifically speaking, their method tried to recursively find a partial permutation matrix for each image based on the assumption that the correspondence of the reordered patterns from preceding rounds of matching had been well established. The patterns from the image being considered were appended in such a way that their rearrangement would make the newly formed matrix highly rank deficient, in which the order of the rearrangement was encoded in the corresponding partial permutation matrix. Their method repeated this process for all images. Although the rank constraint is effective in guiding the search for the partial permutation matrix, there are four shortcomings regarding to this formulation: i) It operates in a recursive mode and only part of the data is used for computing one partial permutation matrix. ii) It assumes that the correspondence from preceding rounds is well established. If this assumption is violated, the whole model would be broken down and the error would be propagated to the subsequent alignment. iii) There is no mechanism to handle noise. iv) It can only handle low dimensional data. In contrast, our proposed method is free from these shortages. This is because it operates in a batch mode, where all the images are considered simultaneously for finding coherent correspondence based on the low-rank constraint. To improve the robustness, a sparse error term is introduced in this work to handle errors such as corruption and occlusion, which are common in images and videos.

Our method is most related to sparse and low-rank matrix decomposition [19]. Their work proves that if a data matrix is superposition of a low-rank matrix and a sparse matrix, each matrix can be exactly recovered. Peng *et al.* [20] extended this framework to robustly align a batch of linearly correlated images despite the gross corruption and occlusion, in which they proposed to find an optimal set of *image domain transformations* (*e.g.*, affine transformations) such that the matrix of the transformed images could be decomposed as the sum of a low-rank matrix of recovered aligned images and a sparse matrix of errors. Our proposed method shares similar spirit but in another direction. It seeks a set of *optimal partial permutation matrices* for correspondence such that the rearranged matrix can be factorized into a low rank matrix and a sparse matrix.

3 Global Correspondence via Sparse and Low-rank Matrix decomposition

In this section, we formulate the correspondence as the search for the optimal partial permutation matrices that minimizes the rank of the reordered features/patches. A sparse error term is also introduced into the framework to improve its robustness to gross corruption and occlusion. Since the proposed optimization problem is not convex, we propose a two-step approach to approximate the solution by solving two subproblems sequentially: minimization of the sum of nuclear norm and l^1 norm for solving the relaxed partial permutation matrices. Then each of the partial permutation matrix is projected to its feasible solution via solving a binary integer programming problem.

3.1 Matrix Rank as a Measure of Correspondence: A Prior

Suppose we are given N images $\mathcal{I}_1, \cdots, \mathcal{I}_N$. For each image $\mathcal{I}_n$, we can extract $\bar{K}$ features $\mathbf{f}_{n,1}, \cdots, \mathbf{f}_{n,\bar{K}} \in \mathbb{R}^d$ at $\bar{K}$ landmark locations, or divide each image into $\bar{K}$ blocks, vectorize the pixel values from the ith block as $\mathbf{f}_{n,i}$ for $i = 1, ..., \bar{K}$ and stack them as a matrix $\mathbf{F}_n = [\mathbf{f}_{n,1}, \cdots, \mathbf{f}_{n,\bar{K}}] \in \mathbb{R}^{d\times\bar{K}}$. As shown in [15, 20], if these features or patches are well aligned (*i.e.,* features from the same landmark location or patches of the same place across different images are put into the well-corresponded entry in the feature space) without noise and outliers, then they should be linearly correlated. Specifically, if we denote vec : $\mathbb{R}^{w\times h} \rightarrow \mathbb{R}^{wh}$ as the operator that stacks a $w \times h$ matrix as a (wh)-dimensional vector, then the matrix $\mathbf{A} = [\text{vec}(\mathbf{F}_1)|\cdots|\text{vec}(\mathbf{F}_N)] \in \mathbb{R}^{d\bar{K}\times N}$ should be approximately low-rank.

3.2 Modeling Correspondence via Partial Permutation Matrix

Now, we consider a more general case where each image $\mathcal{I}_n$ has K_n patterns of size d, and assume that these patterns extracted from N images $\mathbf{F}_1, \cdots, \mathbf{F}_N$ are not well corresponded with respect to each other. Our interest is to find $\bar{K}$ intrinsic patterns for each image where $\bar{K} \leq K_n \forall n \in \{1, \cdots, N\}$, and their correspondence among N images (*i.e.,* N sets of $\bar{K}$ features). Now, we first model the correspondence with partial permutation matrix $\mathbf{P}_n \in \mathcal{P}_n$ for image $\mathcal{I}_n$ similar to [15], where $\mathcal{P}_n$ is defined as follows:

$$\mathcal{P}_n = \{\mathbf{P}_n | \mathbf{P}_n \in \{0,1\}^{K_n\times\bar{K}},\ \mathbf{1}_{K_n}^T\mathbf{P}_n = \mathbf{1}_{\bar{K}}^T,\ \mathbf{P}_n\mathbf{1}_{\bar{K}} \leq \mathbf{1}_{K_n}\},$$

where $\{0,1\}^{K_n\times\bar{K}}$ denotes a $K_n \times \bar{K}$ matrix whose elements are either 0 or 1 and $\mathbf{1}_c$ (*resp.* $\mathbf{0}_c$) denotes a column vector of all 1 (*resp.* 0) of length c. Then, there exist partial permutation matrices $\mathbf{P}_1, \cdots, \mathbf{P}_N$ such that the reordered patterns are well corresponded. In other words, the matrix $[\text{vec}(\mathbf{F}_1\mathbf{P}_1)|\cdots|\text{vec}(\mathbf{F}_N\mathbf{P}_N)] \in \mathbb{R}^{d\bar{K}\times N}$ is rank deficient. Hence the correspondence problem can be formulated as the following optimization problem:

$$\min_{\mathbf{P}_n\in\mathcal{P}_n|_{n=1}^N,\mathbf{L}} \text{rank}(\mathbf{L}), \quad s.t.\ [\text{vec}(\mathbf{F}_1\mathbf{P}_1)|\cdots|\text{vec}(\mathbf{F}_N\mathbf{P}_N)] = \mathbf{L}. \tag{1}$$

3.3 Modeling Noise in Features as Large and Sparse Errors

In practise, error such as corruption and occlusion is common in images and videos, thus the low rank property of the aligned matrix is likely to be violated. To improve the robustness, such error is modeled as a sparse matrix as it only affects a small fraction of the data. Then, the problem (1) is modified as follows:

$$\min_{\mathbf{P}_n \in \mathcal{P}_n|_{n=1}^{N}, \mathbf{L}, \mathbf{E}} \quad \mathrm{rank}(\mathbf{L}) + \lambda \|\mathbf{E}\|_0, \quad s.t.\ [\mathrm{vec}(\mathbf{F}_1\mathbf{P}_1)|\cdots|\mathrm{vec}(\mathbf{F}_N\mathbf{P}_N)] = \mathbf{L} + \mathbf{E}, \tag{2}$$

where $\|\cdot\|_0$ denotes the number of nonzero entries and $\lambda > 0$ is a trade off parameter that balances the rank of the solution and the sparsity of the error.

3.4 Convex Relaxation

The optimization problem is not directly tractable due to the following aspects: i) The minimization on the matrix rank is non-convex; ii) The l^0 norm is hard to optimize; iii) The optimization with respect to $\mathbf{P}_n$ is non-linear, therefore the corresponding integer programming is hard to solve. Moreover, since the rank of a matrix and the l^0 norm are discrete-valued functions, the solution for the above optimization is unlikely to be stable. Meanwhile, it has been shown in [19, 20] that if the rank of the matrix $\mathbf{L}$ to be recovered is not too high and the number of non-zero entries in $\mathbf{E}$ is not too large, minimizing its convex surrogate (*i.e.*, the rank($\cdot$) is replaced with the nuclear norm while the l^0 norm is replaced with the l^1 norm) can exactly recover the low rank matrix $\mathbf{L}$. In addition, it is well known that there is no efficient solution to the general integer programming problem. Similar to [1, 15], we relax the binary constraint to a real value between 0 and 1. Based on the above relaxation and after performing some change of variables,

$$\boldsymbol{\theta}_n = \mathrm{vec}(\mathbf{P}_n), \ \ \mathbf{Z}_n = \begin{bmatrix} \mathbf{I}_{\bar{K}} \otimes \mathbf{F}_n \\ \mathbf{I}_{\bar{K}} \otimes \mathbf{1}_{K_n}^T \end{bmatrix}, \ \ \mathbf{M}_n = \begin{bmatrix} \mathbf{1}_{\bar{K}}^T \otimes \mathbf{I}_{K_n} \\ -\mathbf{I}_{\bar{K}K_n} \end{bmatrix}, \ \ \mathbf{h}_n = \begin{bmatrix} \mathbf{1}_{K_n} \\ \mathbf{0}_{\bar{K}K_n} \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product and $\mathbf{I}_c$ is a $c \times c$ identity matrix, we approximate the solution by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{L}, \mathbf{E}, \boldsymbol{\theta}_1, \cdots \boldsymbol{\theta}_N} \quad & \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1, \\ s.t. \quad & [\mathbf{Z}_1\boldsymbol{\theta}_1, \cdots, \mathbf{Z}_N\boldsymbol{\theta}_N] = \begin{bmatrix} \mathbf{L} + \mathbf{E} \\ \mathbf{1}_{\bar{K}}\mathbf{1}_N^T \end{bmatrix}, \\ & \mathbf{M}_n\boldsymbol{\theta}_n \leq \mathbf{h}_n, \quad \forall n \in \{1, \cdots, N\}, \end{aligned} \tag{3}$$

where $\|\cdot\|_*$ denotes the nuclear norm, and $\leq$ denotes component-wise inequality.

Since the number of variables in the optimization problem (3) is usually large, scalable solution is essential for its practical use. Fortunately, researches have proposed efficient methods for solving the low rank approximation [21, 22]. Motivated by the fast first-order method Alternative Direction Method of

Multiplier (ADMM) [23], we develop a solution to solve the optimization problem (3). For ease of our development that follows, we define the following notations:

$$\mathbf{Q}(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_N) = [\mathbf{Z}_1\boldsymbol{\theta}_1,\cdots,\mathbf{Z}_N\boldsymbol{\theta}_N] - \begin{bmatrix}\mathbf{L}+\mathbf{E}\\ \mathbf{1}_{\bar{K}}\mathbf{1}_N^T\end{bmatrix}, \tag{4}$$

$$I_C(\boldsymbol{\theta}_n) = \begin{cases}0 & \text{if } \mathbf{M}_n\boldsymbol{\theta}_n \leq \mathbf{h}_n\\ +\infty & \text{otherwise}\end{cases}, \tag{5}$$

where $\mathbf{Q}(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_N) = \mathbf{0}$ represents the equality constraint of problem (3), and $I_C(\boldsymbol{\theta}_n)$ is the indicator function of the inequality associated with $\boldsymbol{\theta}_n$ in problem (3). Hence, the augmented Lagrangian function of the optimization problem (3) can be easily written as

$$\begin{aligned}\mathcal{L}_\mu(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_N,\mathbf{Y}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1 + <\mathbf{Y},\mathbf{Q}(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_N)>\\ + \sum_{n=1}^{N} I_C(\boldsymbol{\theta}_n) + \frac{\mu}{2}\|\mathbf{Q}(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_N)\|_F^2, \end{aligned}\tag{6}$$

where $\mathbf{Y} \in \mathbb{R}^{(d+1)\bar{K}\times N}$ is the Lagrange multiplier matrix, μ is a positive scalar, $<.,.>$ denotes the matrix inner product and $\|\cdot\|_F$ denotes the Frobenius norm. An iteration of augmented Lagrangian method [21] for problem (3) are given by

$$(\mathbf{L}^{t+1},\mathbf{E}^{t+1},\boldsymbol{\theta}_1^{t+1},\cdots,\boldsymbol{\theta}_N^{t+1}) := \underset{\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_N}{\arg\min}\ \mathcal{L}_\mu(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_N,\mathbf{Y}^t), \tag{7}$$

$$\mathbf{Y}^{t+1} := \mathbf{Y}^t + \mu\mathbf{Q}(\mathbf{L}^{t+1},\mathbf{E}^{t+1},\boldsymbol{\theta}_1^{t+1},...,\boldsymbol{\theta}_N^{t+1}), \tag{8}$$

where t is the current iteration number, and μ follows the updating rule $\mu^{t+1} = \rho\mu^t$ for some $\rho > 1$ as in [20]. As pointed out in [23], the problem (7) is hard to solve in general, and hence we minimize $\mathcal{L}_\mu(\mathbf{L},\mathbf{E},\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_N,\mathbf{Y}^t)$ with respect to $(\mathbf{L},\mathbf{E})$ and $(\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_N)$ alternatingly, and both of them are relatively easy to solve.

Update L, E: The partial minimization problem of (7) with respect to ($\mathbf{L}$, $\mathbf{E}$) can be equivalently written as follows:

$$(\mathbf{L}^{t+1},\mathbf{E}^{t+1}) := \underset{\mathbf{L},\mathbf{E}}{\arg\min}\ \|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1 + <\bar{\mathbf{Y}}^t,\mathbf{D}^t - \mathbf{L} - \mathbf{E}> + \frac{\mu}{2}\|\mathbf{D}^t - \mathbf{L} - \mathbf{E}\|_F^2, \tag{9}$$

where $\mathbf{D}^t = [\mathbf{F}_1\boldsymbol{\theta}_1^t,\cdots,\mathbf{F}_N\boldsymbol{\theta}_N^t]$, $\bar{\mathbf{Y}}^t \in \mathbb{R}^{d\bar{K}\times N}$ is a matrix containing the first $d\bar{K}$ rows of $\mathbf{Y}^t$. This optimization problem can be solved similarly to [21]:

$$[\mathbf{U},\mathbf{S},\mathbf{V}] = \text{svd}(\mathbf{D}^t - \mathbf{E}^t + \mu^{-1}\bar{\mathbf{Y}}^t),$$

$$\mathbf{L}^{t+1} = \mathbf{U}S_{\mu^{-1}}[\mathbf{S}]\mathbf{V}^T, \tag{10}$$

$$\mathbf{E}^{t+1} = S_{\lambda\mu^{-1}}[\mathbf{D}^t - \mathbf{L}^{t+1} + \mu^{-1}\bar{\mathbf{Y}}^t], \tag{11}$$

where $S_\tau[\mathbf{X}]$ is the shrinkage operator for the matrix $\mathbf{X}$ that applies $S_\tau[x] = \text{sign}(x)\cdot\max\{|x| - \tau, 0\}$ to all the elements of $\mathbf{X}$.

Update $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N$: The partial minimization problem of (7) with respect to $(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N)$ can be decoupled into N independent subproblems, each of which corresponds to $\boldsymbol{\theta}_n$ and can be equivalently formulated as the following convex quadratic programming (QP) problem:

$$\boldsymbol{\theta}_n^{t+1} := \underset{\mathbf{M}_n\boldsymbol{\theta}_n \leq \mathbf{h}_n}{\arg\min} \ \frac{1}{2}\boldsymbol{\theta}_n^T \mathbf{Z}_n^T \mathbf{Z}_n \boldsymbol{\theta}_n + \mathbf{e}_n^T \left(\frac{1}{\mu}\mathbf{Y}^t - \begin{bmatrix} \mathbf{L}^{t+1} + \mathbf{E}^{t+1} \\ \mathbf{1}_{\bar{K}} \mathbf{1}_N^T \end{bmatrix} \right)^T \mathbf{Z}_n \boldsymbol{\theta}_n, \quad (12)$$

where $\mathbf{e}_n$ is a unit column vector with all the entries set to 0 except the n^{th} entry set to 1. This problem can be readily solved by a standard QP solver.

3.5 Converting $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N$ back to binary vectors

Since the solutions $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N$ obtained using the ADMM method above are not necessarily binary vectors, we need to project them back to feasible solutions via the binary integer programming (BIP):

$$\min_{\mathbf{x}_n \in \{0,1\}^{\bar{K}K_n}} -\boldsymbol{\theta}_n^T \mathbf{x}_n, \ \ s.t. \ (\mathbf{I}_{\bar{K}} \otimes \mathbf{1}_{K_n}^T)\mathbf{x}_n = \mathbf{1}_{\bar{K}}, \ \ (\mathbf{1}_{\bar{K}}^T \otimes \mathbf{I}_{K_n})\mathbf{x}_n \leq \mathbf{1}_{K_n}. \quad (13)$$

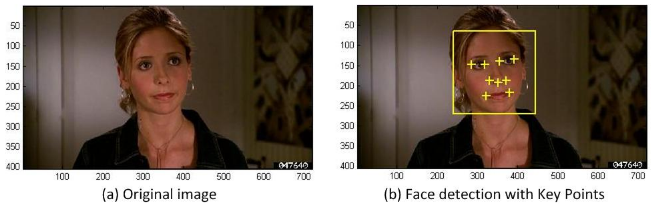
Once the optimal $\mathbf{x}_n$ is acquired, we can recover the permutation matrix. The whole algorithm is summarized in Algorithm 1.

4 Experiments

We evaluate the performance of the proposed method for finding correspondence using only features/patches, in which the spatial information is not taken into consideration. We first quantitatively evaluate the proposed method on finding correspondence among local features extracted from landmarks of a face image, and compare with the state-of-the-art methods such as [15] (referred to as RankConstr) and [16] (referred to as OneShot) that have integrated mechanism to find correspondence for multiple images. In addition, we verify the robustness

Algorithm 1 The optimization algorithm for the proposed method

```
Input: $[\mathbf{F}_1, \cdots, \mathbf{F}_N], \lambda, \rho > 1, \mathbf{Y}^0, \mathbf{E}^0, \mathbf{L}^0, \boldsymbol{\theta}_n^0|_{n=1}^N$
$t = 0$;
WHILE not converge do
    Update $\mathbf{L}^{t+1}$ as in Eqn.(10);
    Update $\mathbf{E}^{t+1}$ as in Eqn.(11);
    Update $\boldsymbol{\theta}_n^{t+1}$ as in Eqn.(12) with QP solver, $\forall n = \{1, \cdots, N\}$;
    Update $\mathbf{Y}^{t+1}$ as in Eqn.(8);
    $t = t + 1$;
End While
Optimize $\mathbf{x}_n$ as in Eqn.(13) with BIP solver, $\forall n = \{1, \cdots, N\}$;
Reshape $\mathbf{x}_n$ back to matrix and store in $\mathbf{P}_n$, $\forall n = \{1, \cdots, N\}$;
Output: $\mathbf{P}_n|_{n=1}^N$
```



(a) Original image

(b) Face detection with Key Points

Fig. 1. A face image extracted from "Buffy the Vampire Slayer". Fig. 1(a) is the original face image. Fig. 1(b) shows the detected face with nine landmarks detected by [25]. The landmarks are the left and right corners of the eyes, the two nostrils and the tip of the nose, as well as the left and right corners of the mouth.

of the competing methods with respect to different levels of corruption and different number of outliers. In the second task, we evaluate the proposed method on face videos from [24], in which there exist variations in pose and expression for the cropped faces. There are two evaluation metrics used in these two sets of experiments: one is the feature level matching precision (referred to as P_{fea}), which measures the percentage of correctly matched patterns with respect to the total number of inlier patterns; The other one is the image level matching precision (referred to as P_{img}), which measures the percentage of perfect matched images over the total number of images. An image is counted as a perfect match if *all* the patterns within the image are well matched to the patterns used in the first image. They are formally defined as follows:

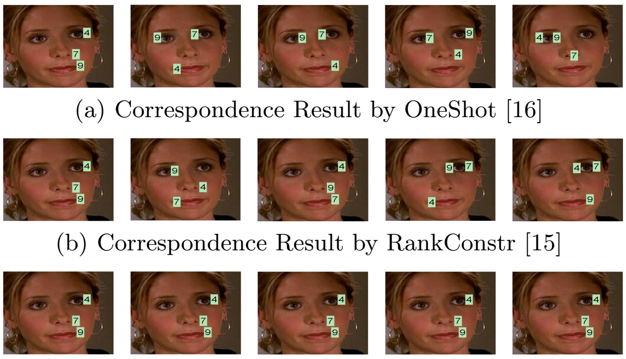
$$P_{fea} = \frac{\text{number of matched patterns}}{\text{total number of inlier patterns}}, \ P_{img} = \frac{\text{number of perfect matched images}}{\text{total number of images}}.$$

In this work, we empirically set $\rho = 1.1$ for both tasks and $\lambda = \frac{1}{3\sqrt{N}}$ (*resp.* $\lambda = \frac{1.3}{\sqrt{dN}}$) for the 1^{st} (*resp.* 2^{nd}) task where d is the number of pixels.

4.1 Features correspondence for landmarks of a face

In this set of experiments, we evaluate the proposed method on finding correspondence among local features extracted from nine landmarks of a face image as shown in Fig. 1. We extract the 128-dimension SIFT feature [26] to form a matrix of size 128 × 9. We randomly rearrange the column order of this matrix for 20 times and stack them as a matrix of size 128 × 180. We are interested in recovering the correspondence from this disordered matrix. We further add white Gaussian noise (*resp.* outliers) to compare the robustness of all the methods.

Results and discussion on noise: There are two factors for generating the white Gaussian noise. One is the percentage of dimensions where the corresponding entry is corrupted. The other is the multiplication factor (denoted as MF) used to define the standard deviation of the white Gaussian noise (*i.e.*, $N(0, \text{MF} \times \max(\text{F})))$, where max(F) is the maximum value from the full data matrix. To reduce the effect of randomness of the pseudo-order and noise, we



(a) Correspondence Result by OneShot [16]

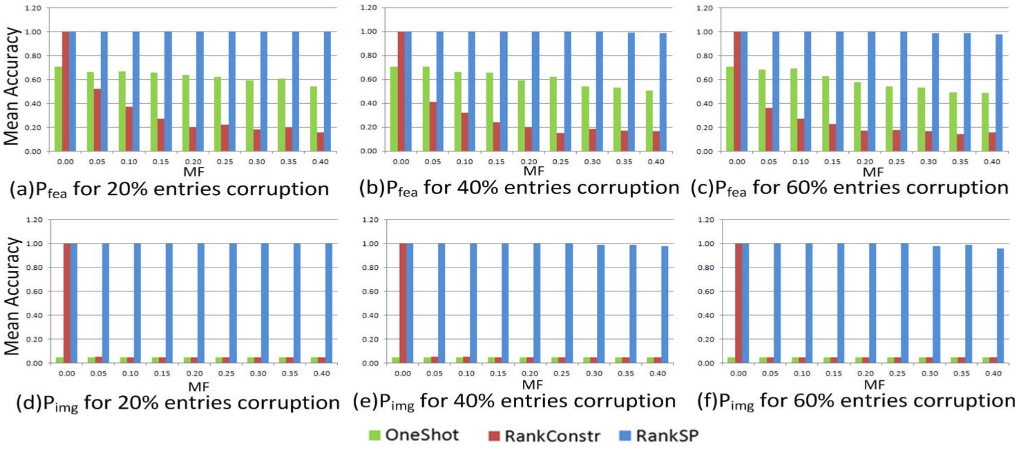
(b) Correspondence Result by RankConstr [15]

(c) Correspondence Result by our proposed RankSP

Fig. 2. An exemplar results using OneShot [16], RankConstr [15] and our proposed RankSP. The results are based on the following experiment setting: 40% of entries are corrupted with multiplication factor (MF) set to 0.2. The features in images from the first column are used as the template for all methods. For better presentation, only the right corner of the right eye (marked as 4), the corner of the right nostril (marked as 7), and the right corner of the mouth (marked as 9) are shown for comparison.

repeat the experiment for 5 times using different orders of random corrupted features and calculate the mean accuracy. An exemplar matching result is shown in Fig. 2 and all the empirical results for different levels of noise are shown in Fig. 3. From these results, we have the following observations:

1. Both RankSP and RankConstr can achieve 100% accuracy for both P_{fea} and P_{img} when there is no noise in the data. This clearly shows the effectiveness to use of the rank constraint for finding correspondence for multiple images.
2. RankConstr is very sensitive to noise, its performance for both evaluation criteria drops significantly even only small amount of corruption is presented in the data. On the other hand, our proposed method RankSP can reliably find the correspondence as the percentage of corrupted entries and the standard derivation of the white Gaussian noise increase. Even when as much as 60% of the entries are corrupted by noise, our proposed method still achieves mean accuracy of 98% (*resp.* 96%) for P_{fea} (*resp.* P_{img}). An explanation is that the proposed method introduces an sparse error term. When there exists error such as corruption in the data, it would be factorized into this error term and therefore the low rank matrix is not severely affected.
3. OneShot has poor mean accuracy on P_{img} using both clean and corrupted data. This is probably because OneShot applies k-means clustering to solve the correspondence problem, in which there is no explicit control to avoid the many-to-one correspondence (*i.e.*, multiple patterns within one image match to one pattern in the template image). Therefore it is very unlikely it can achieve good result on P_{img}. On the other hand, our proposed method RankSP has an integrated mechanism to avoid the many-to-one correspondence by enforcing the constraint of the partial permutation matrix (*i.e.*, one pattern can be at most matched to one pattern in the template).



(a)P_{fea} for 20% entries corruption (b)P_{fea} for 40% entries corruption (c)P_{fea} for 60% entries corruption

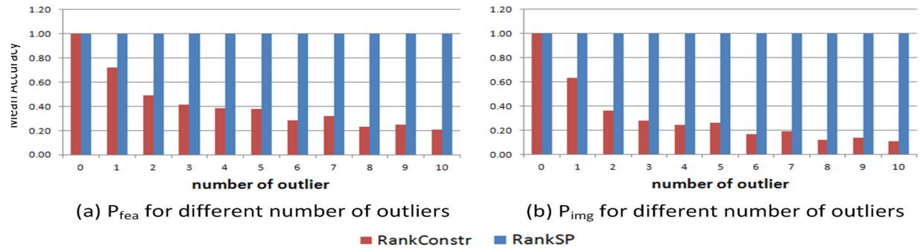
(d)P_{img} for 20% entries corruption (e)P_{img} for 40% entries corruption (f)P_{img} for 60% entries corruption

Fig. 3. Results with respect to different levels of white Gaussian noise. The 1^{st} (*resp.* 2^{nd}) row shows the results for the mean accuracy based on P_{fea} (*resp.* P_{img}).

Results and discussion on outlier: We also investigate the performance variations with respect to the number of outliers. For the above images, we additionally extract SIFT features from random locations. Since OneShot does not have any mechanism to identify outliers, we only compare our proposed method RankSP with RankConstr. Fig. 4 shows the performance variation with respect to the number of outliers. From these experimental results, we find that the proposed method is very robust to the outliers, in which it can reliably achieve 100% mean accuracy using both evaluation criteria as the number of outliers increases from 0 to 10. Meanwhile, the mean accuracy P_{fea} (*resp.* P_{img}) for RankConstr decreases significantly from 100% to around 20% (*resp.* 10%). An explanation is that RankConstr is a greedy approach. It recursively computes a partial permutation matrix for one image at a time using only the reordered features from preceding rounds of calculation (*i.e.*, part of the data is used), in which it assumes that the correspondence among these reordered features has been well established. If this assumption is violated, the error would be propagated to the subsequent matches, leading to many inaccurate matches. In contrast, our method operates in a batch mode and the low rank constraint is enforced on the *full* data matrix, hence it is more robust to the outliers.

4.2 Rearrangement from Dis-order Face Patches in Videos

In this section, we evaluate the proposed algorithm for recovering disordered face videos [24]. Compared to the image used in the previous section, the faces from different frames exist variations in pose and expression. We crop the face from each frame and it is resized to 60 × 60 pixels and divided into 3 × 3 blocks. The order of blocks in each frame are then perturbed independently. In addition, to evaluate the robustness of the proposed algorithm with respect to different levels of occlusion, we generate a uniformly distributed random number between 0 and 1 for each frame. If this number is less than a pre-defined threshold T, then a



(a) P_{fea} for different number of outliers

(b) P_{img} for different number of outliers

Fig. 4. Results with respect to the number of outliers. The left (*resp.* right) figure shows the correspondence results for P_{fea} (*resp.* P_{img}). Since OneShot does not have any mechanism to identify outliers, only results from RankConstr and RankSP are shown in this figure.

randomly selected block from this frame will be filled with black. An exemplar result is shown in Fig. 5 and all the empirical results are shown in Fig. 6. We observe that RankConstr can only handle at most 30 images for this dataset, hence only the first 30 frames for all the videos are used for its performance evaluation. From this set of results, we have the following observations:

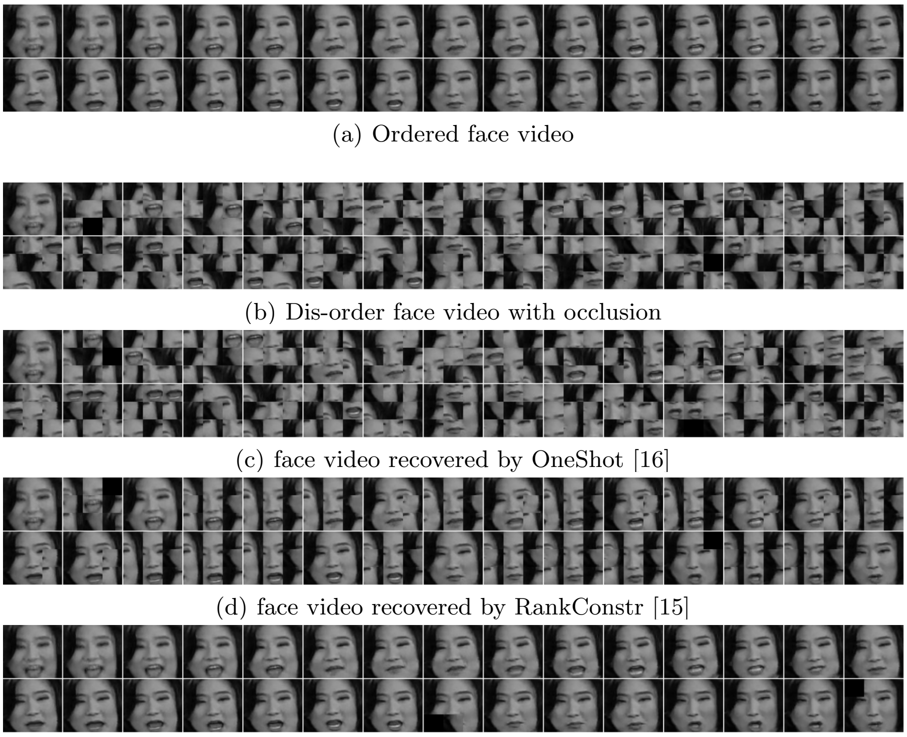
1. Our proposed method RankSP can achieve close to 100% accuracy if only variations in pose and expression exist (*i.e.*, there is no occlusion). An explanation is that the transition in videos is usually smooth, and if there are enough disordered faces from one person to be recovered, the low-rank property would not be significantly affected even if variation in pose exists. In addition, the variation in expression only affects parts of a face (*e.g.*, smile only affects the mouth and cheek areas of a face, frown may only affects the eyebrow area of a face, etc), which is readily handled by the sparse error term in our proposed method. In contrast, OneShot and RankConstr do not have a well-formulated solution to simultaneously handle these two effects, and hence they generally perform worse than our method.
2. When the probability that a block is occluded within a frame increases, the performance of our RankSP does not degrade much, which shows the robustness of the proposed method. Specifically speaking, RankSP can still achieve above 90% for P_{fea} and 80% for P_{img} when the threshold T is set to 0.2 (*i.e.*, approximately 20% of frames are occluded). An explanation is that occlusion only affects small proportion of the face, and such kind of error is readily handled in the proposed method via the sparse error term. In addition, RankSP operates in a batch mode, in which all the images are simultaneously taken into consideration. So even if a misalignment accidentally happens in one frame, it would not adversely affects other frames. In contrast, RankConstr again cannot handle such occlusion. An explanation is that this method lacks of a robust mechanism to handle sparse error and its greedy solution further deteriorates the result.

We also take the Connie Chung video as an example to compare the robustness of our method RankSP with OneShot and RankConstr for 9, 16 and 25 blocks. We report the running time of the two most time-consuming operations

# of blocks	OneShot			RankConstr			RankSP			
	Time(s)	$P_{fea}(\%)$	$P_{img}(\%)$	Time(s)	$P_{fea}(\%)$	$P_{img}(\%)$	Loop(s)	BIP(s)	$P_{fea}(\%)$	$P_{img}(\%)$
9 blocks (3*3)	8	51.7	0.0	393	55.2	20.0	212	6	99.8	99.3
16 blocks (4*4)	39	53.4	0.0	239	50.2	26.7	488	24	93.6	63.0
25 blocks (5*5)	142	53.1	0.0	180	45.6	6.7	1509	52	71.1	14.1

Table 1. Running time and accuracy (P_{fea} and P_{img}) of the Connie Chung video.

(*i.e.*, the while loop and BIP solver) for RankSP together with P_{fea} and P_{img}. The result is shown in Table 1, which is obtained on an IBM workstation with 3.33GHz CPU and 32GB memory. The result shows that RankSP consistently outperforms OneShot and RankConstr as the number of blocks increases. It is also observed that solving the QP problem is the most time-consuming operation inside the WHILE loop. To accelerate our work, one direction is to decompose the large QP into a set of smaller ones by exploring the block-diagonal quadratic term to facilitate parallel computing.



(a) Ordered face video

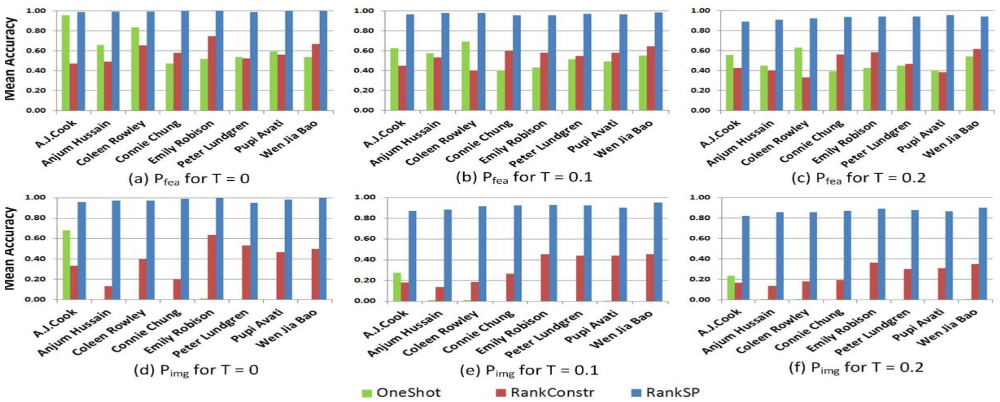
(b) Dis-order face video with occlusion

(c) face video recovered by OneShot [16]

(d) face video recovered by RankConstr [15]

(e) face video recovered by our proposed RankSP

Fig. 5. Sampled cropped face images from a video of Connie Chung [24]. The original video contains 135 frames. Fig. 5(a) shows face images without distortion; Fig. 5(b) shows the corresponding distorted face images with occlusion. Each face image is divided into 3×3 blocks, and the block order within each image is unknown and at most one block per frame may be occluded subjected to threshold (T=0.1 for this figure) except the first image (referred to as a template frame); Fig. 5(c), 5(d) and 5(e) show the result recovered by OneShot, RankConstr and RankSP, repsectively. The comparison clearly shows the effectiveness of our proposed RankSP, in which only one correspondence for a frame in the mid of second row is wrong, while there are gross errors for OneShot and RankConstr.



(a) P_{fea} for T = 0 (b) P_{fea} for T = 0.1 (c) P_{fea} for T = 0.2

(d) P_{img} for T = 0 (e) P_{img} for T = 0.1 (f) P_{img} for T = 0.2

Fig. 6. Face recovering results using OneShot, RankConstr and our RankSP with threshold $T = \{0, 0.1, 0.2\}$. The corresponding number of frames are shown as follows: A.J.Cook(194 frames), Anjum Hussain(71 frames), Coleen Rowley(164 frames), Connie Chung(135 frames), Emily Robison(69 frames), Peter Lundgren(105 frames), Pupi Avati(65 frames), Wen Jia Bao(63 frames).

5 Conclusion

We have proposed a method for finding the correspondence from multiple images by exploiting the priors that the rank of the ordered patterns from a set of images should be lower than disordered patterns, and the error among the reordered pattern is sparse. Based on these priors, we have formulated the correspondence problem as finding a set of optimal partial permutation matrices for the disordered patterns such that the ordered patterns can be factorized as a sum of a low rank matrix and a sparse matrix. A scalable algorithm that solves a sequence of tractable optimization problems has been proposed for the optimal solution, which allows us to efficiently find the correspondence across dozens or even hundreds of images. Moreover, we have shown the efficacy and robustness of our method with extensive experiments using local features extracted from a face with different levels of noise and number of outliers, and the raw pixel values from face videos with variations in pose and expression coupled with occlusion.

Acknowledgement: This study is supported by the Singapore National Research Foundation under its Interactive & Digital Media (IDM) Public Sector R&D Funding Initiative (Grant No. NRF2008IDMIDM004-018) administered by the IDM Programme Office and the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

References

1. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR. (2005)
2. Pollefeys, M., *et al*: Detailed real-time urban 3d reconstruction from video. IJCV **78** (2008) 143–167

3. Enqvist, O., Josephson, K., Kahl, F.: Optimal correspondences from pairwise constraints. In: ICCV. (2009)
4. Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D.A.C.: Graphical models and point pattern matching. TPAMI **28** (2006) 1646–1663
5. Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. TPAMI **25** (2003) 187–199
6. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusion via graph cuts. In: ICCV. (2001)
7. Liu, H., Yan, S.: Common visual pattern discovery via spatially coherent correspondences. In: CVPR. (2010)
8. Torresesani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: ECCV. (2008)
9. Caetano, T.S., McAuley, J., Cheng, L., Le, Q.V., Smola, A.J.: Learning graph matching. TPAMI **31** (2009) 1048–1058
10. Li, H.S., Huang, J.Z., Zhang, S.T., Huang, X.L.: Optimal object matching via convexification and composition. In: ICCV. (2011)
11. Jiang, H., Tian, T.P., Sclaroff, S.: Scale and rotation invariant matching using linearly augmented trees. In: CVPR. (2011)
12. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. TPAMI **29** (2007) 959–975
13. Cho, M., Lee, J., Lee, K.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. In: ICCV. (2009)
14. Barnes, C., Shenchtman, E., Goldman, D.B., Finkelstein, A.: The generalized patchmatch correspondence algorithm. In: ECCV. (2010)
15. Oliveira, R., Costeira, J., Xavier, J.: Optimal point correspondence throught the use of rank constraints. In: CVPR. (2005)
16. Torki, M., Elgammal, A.: One-shot multi-set non-rigid feature-spatial matching. In: CVPR. (2010)
17. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. In: ECCV. (1994)
18. Tomasi, C., Kanade, T.: Shape from motion from image streams under orthography: A factorization method. IJCV **9** (1992) 137–154
19. Candes, E., Li, X., Ma, Y., Wright, J.: Robust principle component analysis? In: Journal of ACM. Volume 58. (2009) 1–37
20. Peng, Y.G., Ganesh, A.: Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In: CVPR. (2010)
21. Lin, Z.C., Chen, M.M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. In: UIUC technical report UILU-ENG-09-2215. (2009)
22. Toh, K.C., Yun, S.W.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. In: Pacific Journal of Optimization. (2010) 615–640
23. Boyd, S., Parikh, N., Chu, E., Peleato, B., Ecjstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. In: Foundations and Trends in Machine Learning. Volume 3. (2010) 1–22
24. Wolf, L., Hassne, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: CVPR. (2011)
25. Everingham, M., Sivic, J., Zisserman, A.: “who are you?” - learning person specific classifiers from video. In: BMVC. (2006)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110