ROBUST VIDEO RESTORATION BY JOINT SPARSE AND LOW RANK MATRIX APPROXIMATION

HUI JI[†], SIBIN HUANG[†], ZUOWEI SHEN[†], AND YUHONG XU[‡]

Abstract. This paper presents a new video restoration scheme based on the joint sparse and low-rank matrix approximation. By grouping similar patches in the spatiotemporal domain, we formulate the video restoration problem as a joint sparse and low-rank matrix approximation problem. The resulted nuclear norm and ℓ_1 norm related minimization problem can also be efficiently solved by many recently developed numerical methods. The efficiency of the proposed video restoration scheme is illustrated on two applications: video denoising in the presence of random-valued noise and video in-painting for archived films. The numerical experiments indicated the proposed video restoration method is compared favorably against many existing algorithms.

Key words. nuclear norm, low-rank matrix, sparse matrix, denoising, in-painting

AMS subject classifications. 68U10, 65J22, 90C25, 65K05

1. Introduction.

Even with today's advances in camera and digital sensor technology, video data collected in practice often suffers from many types of annoying degradations, *e.g.*, noise contamination, image blurring and missing data. For example, video data can be quite noisy when captured at high sensitivities, such as low lighting condition, high ISO setting or high capture rate. The frames of video data can be blurred when there are fast moving objects in the scene or when there are camera shakes. Some parts of video data can be missing or invisible from a user perspective due to occlusions, scratches, or errors in data conversion/communication. The goal of video restoration is then to recover the original one from the obtained degraded video data. With the prevalence of webcams and camera phones, the problem of video restoration has become even more important than before.

In recent years, patch-based image restoration scheme has emerged as one promising approach for various image restoration tasks, e.g. denoising and in-painting ([1, 2, 3]). The basic idea of these approaches is to regularize the restoration process by utilizing the spatial redundancy of original static images. Compared to static image data, video data tends to be of lower quality due to the high speed capturing rate of video camera, e.g. lower signal-to-noise ratio and lower image resolution. Meanwhile, owing to its significant temporal redundancy, video data usually provides much richer information about the scene than static image data does. Thus, the efficiency of restoring degraded video data largely depends on how efficient the temporal redundancy is utilized in the methods. Some of patched based image restoration methods have been extended to the case of video denoising ([4, 5]). Although differing from details, these video restoration methods are built upon the same methodology that exploits the self-similarity in video by examining the the similarity among different image patches.

The aforementioned patch-based image/video restoration methods showed impressive results on suppressing image noise when the noise is mostly Gaussian white noise. In practice, the existence of outliers in the image/video data is not rare and can be caused by many factors, *e.g.* electronic noise and errors in analog-to-digital converter. Also, scratches, blotches and dust are prevalent in archived VHS video

[†] Department of Mathematics, National University of Singapore, Singapore 19076

[‡] Centre for Wavelets, Approximation and Information Processing, National University of Singapore, Singapore 19076

tapes. In the presence of such outliers, the performance of these denoising methods noticeably decreases. Thus, how to develop efficient image/video restoration algorithms that are robust to both image noise and various types of outlier has drawn attentions in recent years (*e.g.* [6, 7, 8]). Most of these robust image/video denoising approaches assume that the outliers can be reliably identified by pre-processing. Such an assumption is valid for some particular type of outliers such as salt-and-pepper noise. For image/video in-painting, most existing methods require the region to be filled are given as an input which usually is done by user interactions. In practice, many types of outlier can not be reliably identified, e.g. random-valued noise. Also, in many situations, the manual identification of outliers many be too time-consuming, e.g. video in-painting.

This paper aims at developing a robust patch-based video restoration algorithm which is capable of simultaneously identifying outliers and recovering the corrupted data. Built upon the same methodology of ”grouping and collaboratively filtering” as many patch-based methods do, the proposed algorithm recovers image patches from the array of similar patches. Motivated by the regularization models proposed in [9, 10] for other applications, we take a similar regularization approach for video restoration that approximating the patch stack by the summation of two components: one is a low-rank matrix that represents clear patches and the other is a sparse matrix that represents outliers. The main advantage of the proposed approach over the one proposed in [8] is outliers do not need to be identified in advance in the proposed approach. Such an advantage makes it very desirable in some video restoration applications including both random-valued noise reduction and video in-painting.

1.1. Related work. There have been an abundant research literature on image/video restoration methods. There are essentially three types of image restoration method: local, non-local and the mixture of them. Our approach is a non-local approach. Thus, we will only discuss the most related non-local techniques. For image noise reduction, the non-local means method ([1]) is one of the pioneering methods to exploit self-similarities in images. It extension to video noise reduction is introduced ([4]) with an accelerated algorithm. In the non-local means approach, the estimated value of each pixel is the weighted average of intensity values of all image pixels in the spatial domain and/or in temporal domain. The weights are determined by the similarity measurements between the patch centered at the pixel being processed and other patches centered at other pixels. The popular BM3D (block matching and 3D filtering) method ([2]) for image denoising takes a similar approach. In the BM3D method, similar image patches are first found and grouped together to form a 3D array, then a shrinkage in 3D transform domain such as wavelet shrinkage or Wiener filter is applied on the 3D array to remove noise. Then the result is obtained by synthesizing a clean image from the de-noised patches. As an extension to BM3D, the VBM3D method ([5]) further leverages the temporal redundancy of video data for better performance. In the VBM3D method, similar patches for the given patch are found within both the image frame and over multiple neighboring image frames. Another type of patch-based image denoising method ([11]) assumes that each image patch has sparse representation under a fixed (unknown) dictionary, i.e. it can be well approximated by the linear combination of a small subset of patches within the dictionary. The restoration, as well as the dictionary, are jointly estimated by minimizing a sparsity-based functional. The sparsity measurement of each patch used in [11] is defined as the number of non-zero coefficients under the dictionary. Such an approach is extended to the case of video de-noising in [12].

Image in-painting problems are first studied by [13]. One representative patch-based approach is the exemplar-based approach proposed by [3], which is based a patch-based greedy sampling method. The exemplar-based approach first selects locations at gap boundary (fill front), then searches and copies matching image patches from known regions to fill the gap. Instead of using existing patches to fill the regions, in [14], the patch to be filled is the one that can be represented by the sparse linear combination of candidate patches under some local patch consistency constraints. the non-local mean approach is also applied for image inpainting ([15]), which used non-local image information from multiple samples within the image. The contribution of each sample to the reconstruction of a target pixel is determined using an weighted similarity function and aggregated to fill the missing information.

Our approach is closely related to the two-stage patch-based methods proposed in [16] and [8]. A so-called LRC (long range correction) method is proposed in [16] to remove impulse noise and scratches from video sequences. In [16], the damaged pixels are first detected by pre-processing. Then for each damaged pixel, the corresponding patch is matched with a damage-free patch of the largest similarity. Then the value of the damaged pixel is replaced by the corresponding pixel in the matched patch. Another two-stage method is proposed in [8] which formulated the problem of removing mixed noise from video sequences to a low-rank matrix completion problem. The type of impulse noise addressed in [8] is salt-and-pepper noise. In [8], image patches with high similarity are first grouped together to form a matrix $P = (p_1, p_2, \ldots, p_n)$ with each column vector representing a vectorized image patch. Since all these matched patches should represent the same scene structure, the un-degraded version of these matched patches should lie in a low dimensional subspace. Based on this observation, the recovery of original image patches is formulated in [8] as the problem of estimating a low-rank matrix L from the given corrupted patch matrix P. When the patch matrix P is only corrupted by Gaussian white noise, the PCA (Principal component Analysis) method can be used to find such a low-rank matrix approximation ([17]). However, in the presence of salt-and-pepper noise, the PCA method is not a suitable one as it is sensitive to outliers. Thus, the sensitivity of the PCA method to outliers is addressed in [8] by estimating the low-rank matrix L only from some reliable elements of the matrix P. In other words, outliers in P are first detected by pre-processing. Then, the low-rank matrix L is estimated from only those elements not belonging to outliers. Such a low-rank matrix estimation problem is the so-called low-rank matrix completion problem which can be solved via convex minimization under certain conditions ([18]). In [8], the recovery of L is done by solving the following minimization problem

$$\min_{L} \|L\|_*, \quad \text{s.t.} \quad \|L|_\Omega - P|_\Omega\|_F \le \delta, \tag{1.1}$$

where Ω denote the index set of the reliable pixels detected by pre-processing, $\|\cdot\|_*$ is the nuclear norm (defined in Section 2.1) and δ is the estimated noise level. There are many efficient methods for solving such a nuclear norm related minimization problem and a fixed point iterative algorithm is used in [8].

1.2. The motivation and our work. The low-rank matrix completion approach proposed in [8] demonstrated an impressive performance on removing mixed noise from video data. It outperformed many existing patch-based algorithms including the VBM3D methd ([5]) and the PCA method ([17]). However, the approach proposed by [8] is only applicable to the applications where the index set Ω is available or can be obtained by pre-processing. As a result, only one particular type of outliers, salt-and-pepper noise, is addressed in [8]. When the salt-and-pepper noise is the only source of outlier, the set Ω can be reliably estimated by pre-processing, such as adaptive median filtering technique ([19, 20]). The requirement on the availability of Ω certainly limits the applicability of the low-rank matrix completion approach ([8]) to other video restoration tasks, e.g. random-valued noise reduction and video in-painting.

The goal of this paper is to overcome the weakness of the approach proposed in [8] by developing a video restoration algorithm that is robust to outliers and does not require the pre-detection of outliers. Built upon the same methodology used in [8, 5, 17], the proposed approach is to restore patch matrix L from the corrupted patch matrix P. Motivated by the recent work on PCP (Principal Component Pursuits) and its applications in background subtraction ([9, 10]), we propose to recover the low-rank patch matrix L from the corrupted patch matrix P by solving either

$$\min_{L,S} \; \|L\|_* + \lambda\|S\|_1, \quad \text{s.t.} \quad P = L + S \tag{1.2}$$

or

$$\min_{L,S} \; \|L\|_* + \lambda\|S\|_1, \quad \text{s.t.} \quad \|P - L - S\|_F \leq \delta \tag{1.3}$$

where L represents the low-rank patch matrix we are seeking for, S represents the outliers, δ is the noise level, λ is a suitable regularization parameter, $\|\cdot\|_*$, $\|\cdot\|_F$ and $\|\cdot\|_1$ are nuclear norm, Frobenious norm and ℓ_1 norm of matrices respectively. It is noted that the outlier matrix S not only contains the outliers of image pixels, but also contains the outliers generated by the mis-matched patch vector. The minimization model (1.2) is to solve the data recovery problems where the data are mostly corrupted by outliers (e.g. video in-painting); and the minimization model (1.3) is to solve the data recovery problems where there also exists other types of random noise besides outliers (e.g. video noise reduction). Two applications are developed in this paper to demonstrate the efficiency of these two minimization models (1.2) and (1.3): one is video de-noising in the presence of random-valued noise; the other is video in-painting for removing artifacts from archived VHS . The remaining of the paper is organized as follows. The section 2 is devoted to the detailed algorithms for robust video restoration, including patch grouping, patch restoration by two minimization models (1.2) and (1.3), and video synthesis from patch matrices. Two related applications and their experimental evaluations, video denoising and video in-painting are given in Section 3. Section 4 concluded the paper.

2. Restoration using joint sparse and low rank matrix approximation.

There are three main components in our algorithms: (i) patches matching and grouping, (ii) recovering low rank patch matrices from the given corrupted matrices and (iii) video synthesis from restored patch matrices.

2.1. Notation.

Before presenting the details of our approach, we first define some notations for the simplicity of discussions. The Frobenius norm and the ℓ_1 norm of a matrix $X \in \mathbb{R}^{n_1 \times n_2}$ are defined by:

$$\|X\|_F = (\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} |x_{i,j}|^2)^{1/2} \quad \text{and} \quad \|X\|_1 = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} |x_{i,j}|$$

respectively, where $x_{i,j}$ is the (i,j)-th entry of X. Assuming that X is of rank r, the singular value decomposition of X with all singular values being non-negative is then defined by

$$X = U\Sigma V^T, \quad \Sigma = \mathrm{diag}(\{\sigma_i\}_{1\leq i\leq r}),$$

where U and V are $n_1 \times r$ and $n_2 \times r$ matrices with orthonormal columns respectively. The nuclear norm of X is defined as the sum of singular values, i.e.

$$\|X\|_* = \sum_{i=1}^{r} |\sigma_i| = \sum_{i=1}^{r} \sigma_i.$$

For each $\tau \geq 0$, let $\mathcal{S}_\tau : \mathbb{R} \to \mathbb{R}$ be the shrinkage operator $\mathcal{S}_\tau(x) = \mathrm{sgn}(x)\max(|x| - \tau, 0)$ and extend it to matrices by applying it element-wisely. The singular value shrinkage operator $\mathcal{D}_\tau(X)$ is then defined as follows ([21])

$$\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^T.$$

It is noted that $\mathcal{S}_\tau(X)$ and $\mathcal{D}_\tau(X)$ are the solutions of the following two minimization problems respectively

$$\min_Y \ \tau\|Y\|_1 + \frac{1}{2}\|Y - X\|_F^2, \quad \min_Y \ \tau\|Y\|_* + \frac{1}{2}\|Y - X\|_F^2.$$

These two shrinkage operator plays an important role in the numerical computation of joint sparse and low rank matrix approximation, as we will see shortly.

2.2. Patch grouping and video synthesis. This subsection focuses on the first and the last components of the algorithm: patch matching and video synthesis from restored patch matrices. Image patch matching is a well-studied problem in image/video processing with a wide range of applications, e.g. motion estimation, tracking and video compression. Given a video sequence $\mathcal{F} = \{f_k\}_{k=1}^K$ with K image frames, we first partition each image f_k into multiple image patches $p_{j,k}$ of size $n \times n$ with overlapping regions. Then, for each image patch $p_{j,k}$, the patches similar to $p_{j,k}$ in all other images and within the neighborhood of this patch need to be found and collected. Such a procedure is called patch matching.

The patch matching problem in our case is slightly different from the standard ones as image patches in our applications could be seriously damaged by noise and outliers. For video in-painting, one may directly apply patch matching algorithm on the raw data as only a small fraction of image pixels are damaged. By using ℓ_1-norm based distance functions as patch similarity measurements, existing un-damaged image pixels usually provide sufficient information to yield satisfactory matching results with few mis-matched patches. Those few mis-matched patches can be viewed as outliers in the patch stack and will be handled by the proposed restoration scheme (more details is given in Section 2.3). In the presence of significant impulse noise (e.g. 30% pixels are completely damaged), directly applying patch matching algorithms on such noisy data is not suitable as the similarity measurements are seriously distorted by those damaged pixels. It is demonstrated in [8] that the performance of patch matching will seriously degrade in the presence of significant pepper-and salt noise. Similar to [8], the patch matching algorithm is not directly carried on the raw video data, but on the data pre-filtered by some median filter. This will produce much more accurate patch matching results than directly using raw data does. It is noted that the pre-filtered data is only for the purpose of patch matching. The raw data is used as the input of the de-noising module.

One main concern in patch matching for video sequences is the computational efficiency. Given a reference patch, exhaustive search for similar patches in the full spatiotemporal domain could be very time consuming. There have been extensive research works on fast patch matching algorithms, especially for motion estimation in video compression. In this paper, we implemented a modified version of the fast three-step hierarchical search algorithm developed in [22] for its implementation simplicity and computational efficiency. Interesting readers are referred to [22] for more details.

Once the corrupted image patches are effectively restored by the algorithms described in Section 2.3. The last step of our algorithm is to seamlessly integrate the restored patches to restored image frames. In our implementation, the image patches are sampled with overlapping regions. Thus, each pixel in image frames might be covered by several restored patches. Similar to most patch-based methods, in our synthesis process, the intensity value of each pixel is determined by taking the average of all estimates from related image patches, which will help smoothing out the possible discontinuity artifacts along the boundaries of patches.

2.3. Joint sparse and low rank matrix approximation. For each reference patch $\mathbf{p}$, similar patches are found in the spatiotemporal domain by using the patch matching algorithm described in the previous section. Assume that m match patches are found and denoted as $\{\mathbf{p}_i\}_{i=1}^m$. If each patch $\mathbf{p}_i$ is represented as a vector $p_i \in \mathbb{R}^{n^2}$ by concatenating all columns of the patch into a column vector, the resulting patch stack is then a $n^2 \times m$ matrix P defined as follows,

$$P = (p_1, p_2, \ldots, p_m).$$

As the matrix P can be corrupted by random noise, outliers or both of them, P can then be decomposed as the summation of three terms:

$$P = L + N + S, \tag{2.1}$$

where L is the original patch matrix for recovery, N is the random image noise and S is the matrix of outliers.

The next goal is to recover L from the matrix P. Recall that L is supposed to be the collection of all matched patches that represent similar contents, thus the rank of L should be low. The matrix S represents the outliers. It is reasonable to assume that the percentage of outliers is not large in the observation P. Thus, the number of non-zero elements of S is small, i.e., S is a sparse matrix. Based on these two observations, we propose to jointly estimate L and S by solving the following minimization when there is little noise (N is close to 0):

$$\min_{L,S} \ \|L\|_* + \lambda\|S\|_1, \quad \text{s.t.} \quad P = L + S, \tag{2.2}$$

or the following minimization when there is noticeable noise:

$$\min_{L,S} \ \|L\|_* + \lambda\|S\|_1, \quad \text{s.t.} \quad \|P - L - S\|_F \leq \delta, \tag{2.3}$$

where λ is some positive regularization parameter and δ is the standard deviation of random noise N.

The two minimization models (2.2) and (2.3) above have been proposed in [9, 10] to extract low-dimensional structure from a grossly corrupted and possibly noisy data matrix. It could be viewed as a replacement of the Principal Component Analysis

(PCA) method for better robustness to outliers. These two minimization approaches are termed as Principal Component Pursuit (PCP), with its robustness to outliers demonstrated in [9] for solving the problem of background subtraction in video surveillance. In their approach, the observed video matrix (array of image frames) is decomposed into the low-rank matrix structure (static background) and the sparse matrix structure (moving objects). It is shown in [9] that under some milder conditions, the minimizers of (2.2) will give an exact recovery of L and S given $N = 0$ and a suitable value of λ. For (2.3), it is shown in [10] that the resulted minimizer gives stable estimates of L and S, i.e., the solutions obtained from (2.3) are close to the true solutions L and S.

In our approach, instead of directly solving (2.3), we solve its penalized form:

$$\min_{L,S} \; \|L\|_* + \lambda\|S\|_1 + \frac{1}{2\mu}\|P - L - S\|_F^2 \tag{2.4}$$

for some suitable value of μ. The performance of the regularization approach (2.4) is highly dependent on the value of two parameters λ and μ. In our approach, the value of λ is set the same as [9] suggested:

$$\lambda = 1/\sqrt{\max(n_1, n_2)}.$$

where n_1, n_2 are the number of rows and columns of the matrix. Regarding the value of μ, we follow the empirical formula proposed in [23] by setting

$$\mu = (\sqrt{n_1} + \sqrt{n_2})\sigma,$$

where σ is the standard deviation of image noise N.

The remaining of the section is devoted to the numerical algorithms of solving (2.2) and (2.4). In recent years, there have been great progresses on how to efficiently solve ℓ_1 norm related minimization problem. One is the so-called accelerated proximal gradient (APG) method has been a popular method with theoretically justified fast convergence rate. Also, it shows impressive performance on solving various ℓ_1 norm related minimization problems arising from imaging applications (*e.g.* [24, 25]) and nuclear norm related matrix completion/decomposition problems ([26, 27]). The convergence of the APG method is guaranteed if suitable parameters are chosen; interested readers are referred to [24] for more details regarding theoretical analysis. Another promising approach is the ADMM (alternating directions method of multipliers) studied in [28, 29, 30]) which also can efficiently solve such problems. In our approach, we used the APG method to solve the minimization problems (2.2) and (2.4).

Algorithm 1 describes the general APG scheme for solving the following unconstrained minimization problem:

$$\min_{X} \; g(X) + f(X), \tag{2.5}$$

where g is a non-smooth function, f is a smooth function and the constant L_f in Algorithm 1 denotes the Lipschitz constant of the gradient of f. The minimization problem (2.4) can be converted to (2.5) by setting

$$\begin{cases} X &= (S, L), \\ g(X) &= \mu\|L\|_* + \lambda\mu\|S\|_1, \\ f(X) &= \frac{1}{2}\|P - L - S\|_F^2. \end{cases} \tag{2.6}$$

When applying Algorithm 1 to solve (2.4), the minimization problem in Step 4 of Algorithm 1 becomes (noticing $L_f = 2$ in our case)

$$\min_{L,S} \ \mu\|L\|_* + \lambda\mu\|S\|_1 + \|L - G_k^L\|_F^2 + \|S - G_k^S\|_F^2.$$

Since L and S are separable in the above minimization, the solutions L and S can be obtained separately by applying singular value shrinkage operator on G_k^L and soft-shrinkage operator on G_k^S, i.e.

$$L_{k+1} = \mathcal{D}_{\mu/2}(G_k^L), \quad S_{k+1} = \mathcal{S}_{\lambda\mu/2}(G_k^S).$$

The detailed algorithm for solving (2.4) is described in Algorithm 2.

Algorithm 2 can be modified to find an approximate solution to (2.2). Notice that the minimization problem (2.2) is equivalent to (2.4) by letting $\mu \to 0$. Thus, we still apply Algorithm 2 to solve (2.2) but the value of μ at each iteration of Algorithm 2 is decreasing until it reaches a pre-defined small positive number $\bar{\mu}$. In other words, $\mu_k = \max(\rho^k\mu, \bar{\mu})$ for $k = 1, 2, \ldots$. The detailed algorithm is given by Algorithm 3, whose output will give a good approximate solution to (2.2).

Algorithm 1 general Accelerated Proximal Gradient method

```
1. do
2.     $Y_k = X_k + \frac{t_{k-1}-1}{t_k}(X_k - X_{k-1})$;
3.     $G_k = Y_k - \frac{1}{L_f}\nabla f(Y_k)$;
4.     $X_{k+1} = \arg\min_X \ g(X) + \frac{L_f}{2}\|X - G_k\|_F^2$;
5.     $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$, $k \leftarrow k+1$;
6. until converged
```

Algorithm 2 Accelerated Proximal Gradient method for solving (2.4)

```
$L_0 = L_{-1} = 0$; $S_0 = S_{-1} = 0$; $t_0 = t_{-1} = 1$;
do
    $Y_k^L = L_k + \frac{t_{k-1}-1}{t_k}(L_k - L_{k-1})$, $Y_k^S = S_k + \frac{t_{k-1}-1}{t_k}(S_k - S_{k-1})$;
    $G_k^L = Y_k^L - \frac{1}{2}(Y_k^L + Y_k^S - P)$, $G_k^S = Y_k^S - \frac{1}{2}(Y_k^L + Y_k^S - P)$;
    $(U, \Sigma, V) = svd(G_k^L)$, $L_{k+1} = U\mathcal{S}_{\mu/2}(\Sigma)V^T$;
    $S_{k+1} = \mathcal{S}_{\lambda\mu/2}(G_k^S)$;
    $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$, $k+1 \leftarrow k$;
until converged
```

3. Applications in video restoration. Based on the algorithms described in Section 2, two applications are developed in this section: one is removing random-valued noise from video data; the other is restoring video data corrupted by artifacts such as hairs, scratches and blotches, which are often seen in archived early films and video tapes. Both video data used for simulation[1] and real degraded video data[2] are available online. Through all experiments in this section, by default, $K = 50$ image frames are used, the patch size is set as 8×8 pixels and sampled with sample rate

[1] http://media.xiph.org/video/derf

[2] http://www.archive.org

Algorithm 3 Accelerated Proximal Gradient method for solving (2.2)

```
$L_0 = L_{-1} = 0;\ S_0 = S_{-1} = 0;\ t_0 = t_{-1} = 1;\ \mu_0 > \bar{\mu} > 0, \rho < 1;$
*do*
    $Y_k^L = L_k + \frac{t_{k-1}-1}{t_k}(L_k - L_{k-1}),\ Y_k^S = S_k + \frac{t_{k-1}-1}{t_k}(S_k - S_{k-1});$
    $G_k^L = Y_k^L - \frac{1}{2}(Y_k^L + Y_k^S - P),\ G_k^S = Y_k^S - \frac{1}{2}(Y_k^L + Y_k^S - P);$
    $(U, \Sigma, V) = svd(G_k^L),\ L_{k+1} = U\mathcal{S}_{\mu_k/2}(\Sigma)V^T;$
    $S_{k+1} = \mathcal{S}_{\lambda\mu_k/2}(G_k^S);$
    $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2};\ \mu_{k+1} = \max(\rho\mu_k, \bar{\mu});\ k+1 \leftarrow k;$
*until converged*
```

4 pixels each along both axes. For each reference patch, 5 patches with the highest matching score in each image frame are selected. Thus, totally 250 patches are stacked for the reference patch in our experiments, which leads to the size of each patch matrix used in our algorithm being 64×250. The default value of ρ in Algorithm 3 is $5/8$ and the default value of $\bar{\mu} = 10^{-6}\mu$. For the purpose of computational efficiency, the maximum number of the iterations of Algorithm 2 and Algorithm 3 is set as 20.

3.1. Video denoising. There are many type of image noise sources for image data in practice ([31]). Some main types of image noise include amplifier noise (Gaussian noise), photon shot noise, impulse noise and quantization noise. Impulse noise often arises in data acquisition and transmission due to faulty sensor or transmission errors ([32]). For example, as a result of electronic noise, there will be "snow" (random dot pattern) seen in video and television with poor (analog) television reception or on VHS tapes. How to remove impulse noise is one important problem in image/video restoration. There are two main types of impulse noise: salt-and-pepper noise and random-valued impulse noise. The definitions of these two types of impulse noise are given as follows. Let x_{ij} and $S_p(x_{i,j})$ denote the original intensity value of a given pixel and the value corrupted by impulse noise respectively. Let the dynamic range of intensity value is $[d_{min},\ d_{max}]$ ($d_{min} = 0, d_{max} = 255$ in our experiments).

- *Salt-and-pepper noise:* a certain proportion of pixels are altered to be either d_{min} or d_{max}, i.e

  $$S_p(x_{ij}) = \begin{cases} d_{min}, & \text{with probability } s/2 \\ d_{max}, & \text{with probability } s/2 \\ x_{ij}, & \text{with probability } (1-s) \end{cases}$$

  where s is the level of salt-and-pepper noise.

- *Random valued impulse noise:* a certain proportion of pixels are altered to be a (uniform) random number in $[d_{min},\ d_{max}]$

  $$S_p(x_{ij}) = \begin{cases} d_{ij}, & \text{with probability } r \\ x_{ij}, & \text{with probability } (1-r) \end{cases}$$

  where d_{ij} is a uniformly distribution random number in $[d_{min},\ d_{max}]$ and r is the level of random-valued noise.

As salt-and-pepper noise only takes two extreme values, it is much easier to detect than random-valued noise does. Thus, in [8], salt-and-pepper noise is first detected by the adaptive median filters ([19]). Then the damaged pixels are explicitly discarded and the problem of image denoising becomes an image inpainting problem. Good results are reported in [8]. However, such a two-stage approach does not work for random-valued noise as it is very hard to be accurately detected and to be removed thereafter. As Gaussian noise is the most prevailing type of image noise, we assume that the video data is corrupted by both Gaussian noise and random-valued impulse noise.

Different from the two-stage methods, our proposed approach simultaneously identify and remove outliers. In our approach, the random-valued impulse noise is viewed as outliers. Then Algorithm 2 can be universally used to remove either salt-and-pepper noise or random-valued noise, as Algorithm 2 does not require the prior knowledge of the locations of outliers. In the experiments, we only focus on the performance of removing random-valued noise as the performance of existing random-valued noise removers is not satisfactory in practice.

The performance of the proposed algorithm in terms of the PSNR value is illustrated in Table 3.1 with respect to different noise levels. The experiment is carried on the "mobile" video sequence with one key frame shown in Fig. 3.1 (a). The image noise is mixed by both Gaussian white noise and random-valued impulse noise. The standard deviation of Gaussian noise σ varies from 5 to 40, and the percentage of pixels corrupted by impulse noise r varies from 10% to 50%. The average PSNR values of one de-noised image over 5 trials are given in Table 3.1. It is seen that the performance decreases more rapidly with the increasing of random-valued impulse noise level than that with the increasing of Gaussian noise level. In other words, random-valued impulse noise is more difficult to remove than Gaussian noise does. The observation is not surprising as the information of the original intensity value of the pixel is completely lost when corrupted by random-valued noise, but partial information is still kept when the pixel is corrupted by Gaussian white noise.

Our approach is compared against four related patch-based video denoising algorithms: the VBM3D (video block matching and 3D filtering) method by [5], the LRC (long range correction) method by [16], the PCA (Principal Component Analysis) method by [17] and the MC (low-rank matrix completion) method by [8]. Since the VBM3D method and the PCA method do not have built-in mechanisms to handle the outliers. The results are quite poor when directly applying them to remove mixed noise from data. Thus, we first apply the widely used center weighted adaptive median filter (see [20] for more details) on the given video data to remove outliers, then feed the pre-processed data to two methods for de-noising. In the experiments, the VBM3D method uses its own internal patch matching procedure. The LRC method, the PCA method, the MC method and the proposed algorithms use the same patch matching algorithm described in Section 2.2. The data used in the patch matching algorithm is the data pre-filtered by the center weighted adaptive median filter.

Three video sequences are used for the experimental evaluation with both Gaussian white noise and random-valued impulse noise added. Table 3.2 listed the average PSNR values over 5 trials of the results from all listed methods. Clearly, the proposed method outperformed all other methods in most tested data sets with noticeable gains in terms of PSNR value. The same conclusion also holds in terms of visual quality. The original clean data, corrupted one and the recovered results from all five methods are shown in Fig. 3.1 – Fig. 3.3 (d)–(h) respectively. It is seen that the results from the VBM3D method, the LRC method and the PCA method showed severe distortions, especially in the cluttered areas. The results from MC method is the second best with

$\sigma \setminus$ r		10%	20%	30%	40%	50%
5	input	17.73	14.76	13.01	11.78	10.81
	output	32.32	28.80	24.27	19.19	15.63
10	input	17.47	14.64	12.95	11.73	10.78
	output	29.69	26.99	23.44	19.59	16.05
20	input	16.63	14.22	12.70	11.57	10.68
	output	26.14	24.12	21.76	18.87	16.29
30	input	15.58	13.67	12.35	11.33	10.51
	output	23.78	21.96	20.09	18.02	16.08
40	input	14.54	13.04	11.93	11.05	10.32
	output	21.87	20.15	18.56	16.96	15.54

TABLE 3.1

Average PSNR value (dB) of tested noisy data and the denoised results over five trials using Algorithm 2.

less artifacts, as there are good built-in mechanisms to cope with outlier. Overall, our algorithm yielded the most visually pleasant results which are also validated in terms of their PSNR values.

It is not surprising to see in most cases, the proposed algorithm significantly outperformed the low-rank matrix completion algorithm on removing random-valued noise from video data. Although the methodology of both algorithms is the same: recover the low-rank structure from given patch matrices, their mechanisms to handle outliers are different. The low-rank matrix completion algorithm need to know the locations of outliers in order to avoid them during the de-noising. Thus, its performance is highly dependent on the accuracy of outlier identification. While salt-and-pepper noise can be accurately detected by adaptive median filter, it is not the case for random-valued noise. It is observed that the pixels around image edges are likely to be falsely identified as outliers by the median filtering and discarded in the recovery of low-rank matrix. As a result, the results from the matrix completion algorithm tend to lose certain image details such as text content in the image frames. On the contrary, the proposed algorithm introduced a sparse matrix that represents the outliers such that the outliers are simultaneously detected when estimating the low-rank matrix. Thus, the proposed algorithm does not suffer from such issues and yield better results than that from the low-rank matrix completion algorithm as shown in Fig. 3.1 – Fig. 3.3.

3.2. Video in-painting. In this section, we apply Algorithm 3 described in Section 2 to repair videos corrupted by line scratches, hairs and dust by viewing these artifacts as outliers. Such artifacts often arise from archived VHS tapes with various causes such as aging, chemical decomposition, improper storage and handling (see e.g. [33, 34, 35]). For example, dust and thin hairs are mainly caused by improper storage and handling, and the line scratch artifact often appears as a vertical bright or dark line which is caused by film abrasion. Both simulated data and real data are tested in the experiments. The simulated video clips are synthesized by applying the so-called old film effect to the clean videos by using professional video edit software Sony Vegas 7.0. Two types of damages are simulated in the experiments: hairs and scratches. It is noted that usually only a fairly small number of pixels are damaged and other pixels are free of noise or the noise level is very low. So, in this experiment, we can directly perform patch matching algorithm on the raw data without applying

Video	(σ, r)	Noisy	VBM3D	LRC	PCA	MC	Alg. 2
		frame	[5]	[16]	[17]	[8]	
'Mobile'	$(5, 10\%)$	18.01	23.70	22.91	23.57	24.10	**32.32**
	$(10, 20\%)$	14.95	22.41	21.20	22.09	23.01	**26.99**
	$(20, 30\%)$	12.99	20.72	19.40	20.08	21.24	**21.76**
'Bus'	$(5, 10\%)$	17.72	26.50	25.07	26.51	25.05	**31.97**
	$(10, 20\%)$	14.68	24.37	22.87	24.54	24.26	**28.41**
	$(20, 30\%)$	12.68	22.44	20.22	22.68	22.74	**23.25**
'Templete'	$(5, 10\%)$	17.59	28.74	27.54	28.49	29.17	**34.19**
	$(10, 20\%)$	14.53	27.13	25.53	26.66	28.21	**29.76**
	$(20, 30\%)$	12.57	24.40	23.02	24.85	**25.67**	24.12

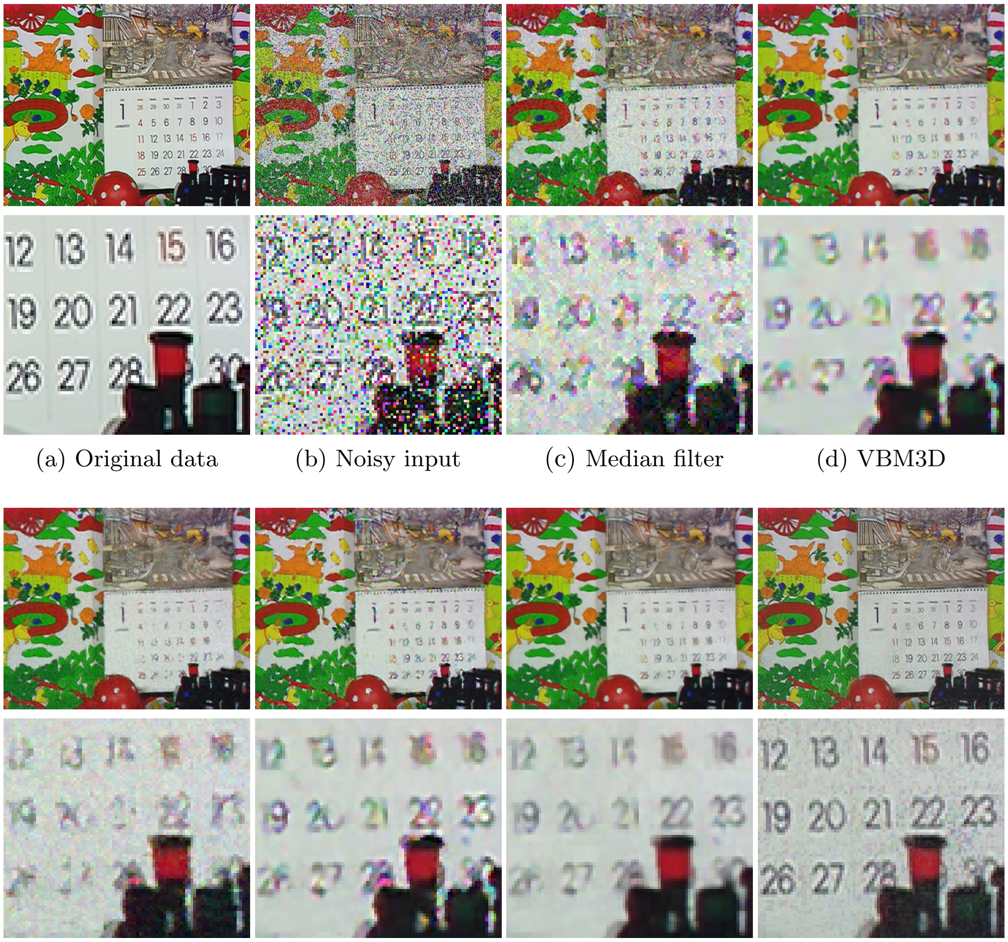
Table 3.2

Average PSNR value (dB) of the results over five trials for video denoising in the presence of random-valued impulse noise with percentage r and Gaussian noise with standard deviation σ.

pre-processing. As the noise level of the pixels except those outliers is very low, we choose the minimization model (2.2) and use Algorithm 3 in the stage of restoring patch matrices.

The proposed Algorithm 3 is evaluated on two synthesized video sequences and two real video sequences with the comparison to the approach of the LRC method, the VBM3D method, the PCA method and the MC method. The VBM3D method and PCA method do not have built-in module to handle the outliers, the scratches and hairs, so we used the data pre-filtered by the adaptive median filter as the input of these two methods. The LRC method and the MC method are both the two-stage method that can re-estimate the "bad" pixels detected by pre-processing. Thus, we use the adaptive median filter to detect the pixels damaged by those artifacts and apply the LRC method and the MC method to in-paint these damaged pixels. On the contrary, Algorithm 3 do not need any pre-detection process and is run directly on the raw data. The visual results are shown in Fig. 3.4–3.5 for the synthesized sequences and Fig. 3.6–3.7 for two real data sequences. It is seen from the experiments on synthesized data that the LRC method, the VBM3D method, the PCA method and the MC method, tend to either over-smoothed images with blurring effect in some sequences or fails to fixed the scratches in other sequences. On the contrary, the performance of Algorithm 3 is consistent on all tested sequences. The results from Algorithm 3 not only removed most artifacts, but also kept most details. For the experiments on real data, it is seen that all other methods more or less kept some noticeable artifacts un-removed while Algorithm 3 removed most artifacts. Overall, Algorithm 3 is capable of removing those artifacts, hairs and scratches, and still preserve image details and sharpness. The main advantage of our video in-painting algorithm over existing algorithms lies in the simultaneous identification and restoration of damaged regions, while other need to employ separate detection schemes to detected damaged areas before in-painting whose accuracy cannot be guaranteed in practice. As a result, the applicability of our approaches is much broader and does not need any prior knowledge on the image damage pattern as others do. For transient features in the video sequence, they are also treated as outliers and thus be removed in our approach.

4. Conclusion. In this paper, we develop a new video restoration scheme based on the joint sparse and low-rank matrix approximation. Built upon the same group-and-filter methodology used in many state-of-art video denoising methods, we convert



(a) Original data (b) Noisy input (c) Median filter (d) VBM3D

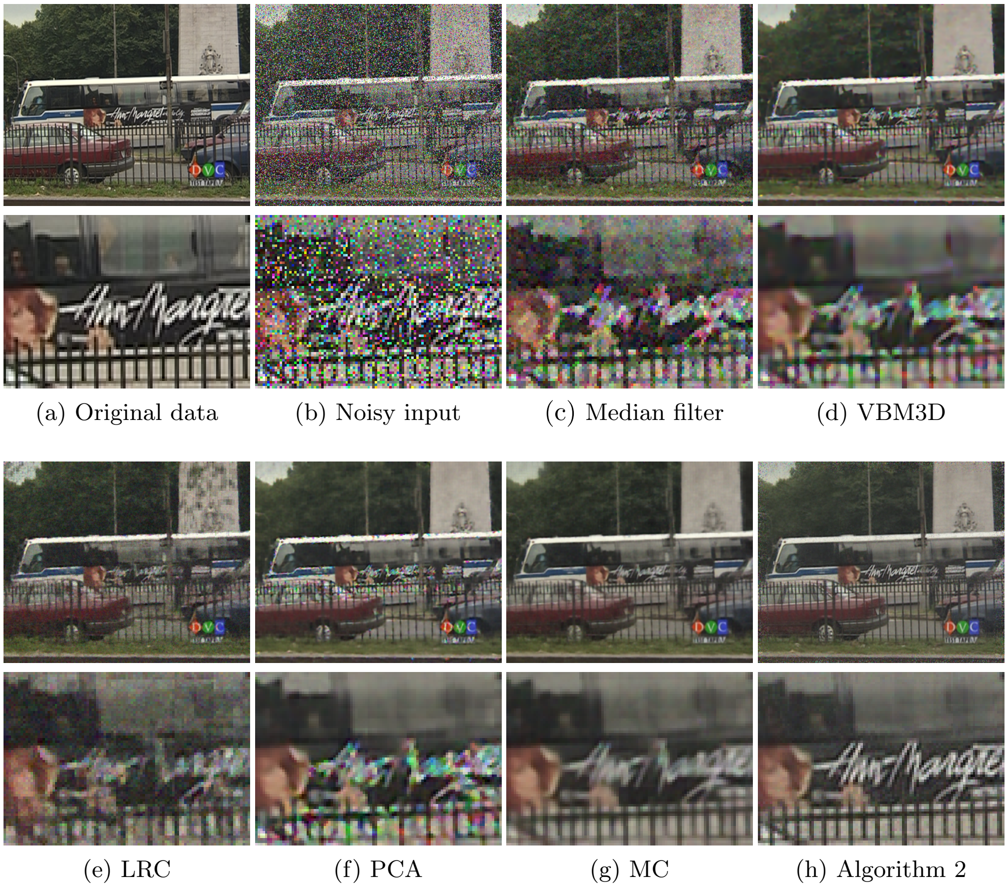
(e) LRC (f) PCA (g) MC (h) Algorithm 2

Fig. 3.1. *Video de-noising: the image frames ares shown in the first row with their magnified versions shown in the second row. (a) the un-corrupted data; (b) the one corrupted by Gaussian noise ($\sigma = 20$) and 30% random-valued impulse noise; (c)–(h): the result from the adaptive median filtering; the VBM3D method ([5]); the LRC method ([16]); the PCA method ([17]); the MC method ([8]); and Algorithm 2 respectively.*

the video restoration problem to a joint sparse and low rank matrix approximation problem. The resulted ℓ_1 norm related minimization problem can also be efficiently solved by many recently developed numerical methods. The proposed video restoration scheme is used to solve two video restoration problems: video denoising in the presence of random-valued noise and video in-painting for archived films. The numerical experiments showed the proposed video restoration method is compared favorably against many existing algorithms, which justified the advantages of the proposed integrated approach over those two-stage methods.

REFERENCES

[1] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithm, with a new one. *SIAM Multiscale Modeling and Simulation*, 4(2):490–530, 2005.



(a) Original data (b) Noisy input (c) Median filter (d) VBM3D

(e) LRC (f) PCA (g) MC (h) Algorithm 2

Fig. 3.2. *Video de-noising: the image frames ares shown in the first row with their magnified versions shown in the second row. (a) the un-corrupted data; (b) the one corrupted by Gaussian noise ($\sigma = 20$) and 30% random-valued impulse noise; (c)–(h): the result from the adaptive median filtering; the VBM3D method ([5]); the LRC method ([16]); the PCA method ([17]); the MC method ([8]); and Algorithm 2 respectively.*

[2] K. Dabov, V. Katkovnik R. Foi, K. Egiazarian, and S. Member. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007.

[3] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Tran. Image Processing*, 13(9):1200–1212, 2004.

[4] M. Mahmoudi and G. Sapiro. Fast image and video denoising via non-local means of similar neighborhoods. *IEEE signal processing letters*, 12(12):839–842, 2005.

[5] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In *Proc. 15th European Signal Processing Conference*, 2007.

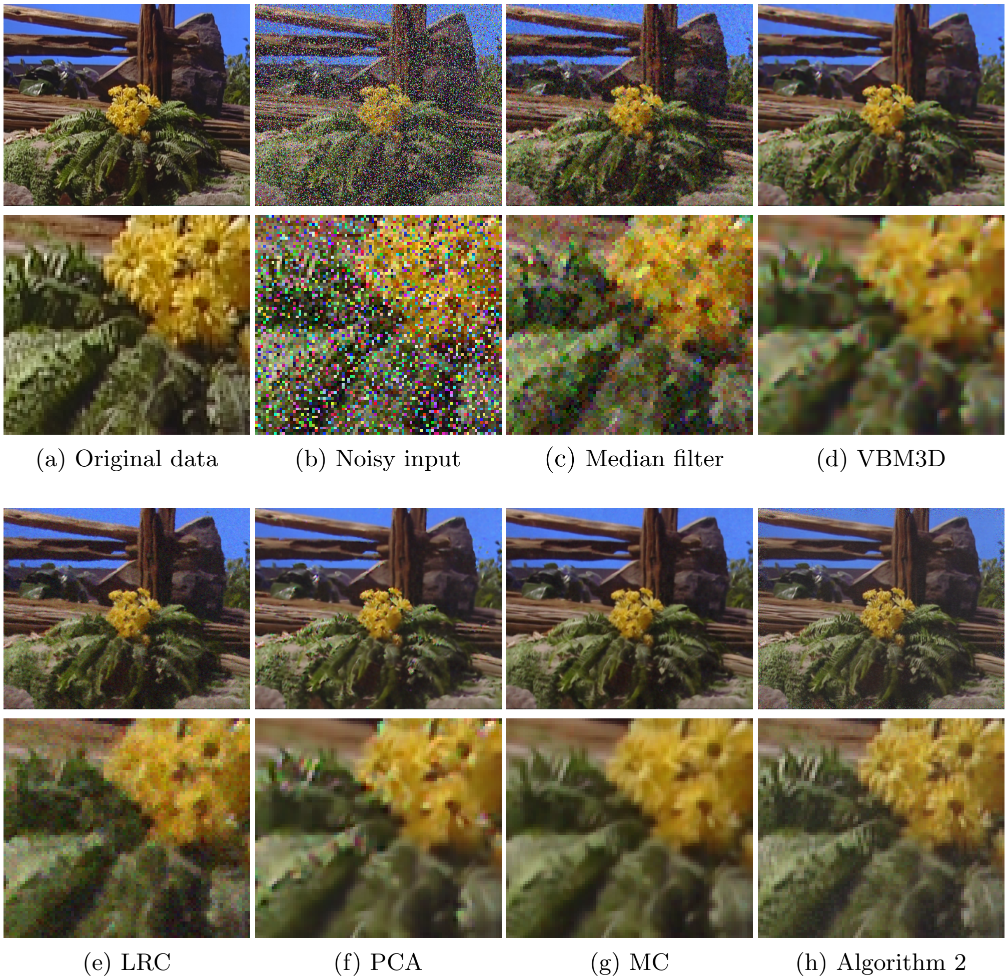
[6] R. H. Chan, C. W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detector and edge preserving regularization. *IEEE Trans. Image Processing*, 14(10):1479–1485, 2005.

[7] Y. Xiao, T. Zeng, J. Yu, and M. Ng. Restoration of images corrupted by mixed gaussian-impulse noise via l1-l0 minimization. Technical Report 10-05, HKBU, 2010.

[8] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust matrix completion by using low rank matrix completion. In *IEEE CVPR*, 2010.

[9] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? Technical report, Stanford University, 2010. preprint, 2009.

[10] Z. Zhou, X. Li, J. Wright, E. J. Candes, and Y. Ma. Stable principal component pursuit. In



(a) Original data (b) Noisy input (c) Median filter (d) VBM3D

(e) LRC (f) PCA (g) MC (h) Algorithm 2

Fig. 3.3. *Video de-noising: the image frames ares shown in the first row with their magnified versions shown in the second row. (a) the un-corrupted data; (b) the one corrupted by Gaussian noise ($\sigma = 20$) and 30% random-valued impulse noise; (c)–(h): the result from the adaptive median filtering; the VBM3D method ([5]); the LRC method ([16]); the PCA method ([17]); the MC method ([8]); and Algorithm 2 respectively.*

Proceedings of IEEE International Symposium on Information Technology (ISIT), 2010.

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12), 2006.

[12] M. Protter and M. Elad. Image sequence denoising via sparse and redundant representations. *IEEE Trans. Image Processing*, 18(1):27–36, 2009.

[13] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, pages 417–424, 2000.

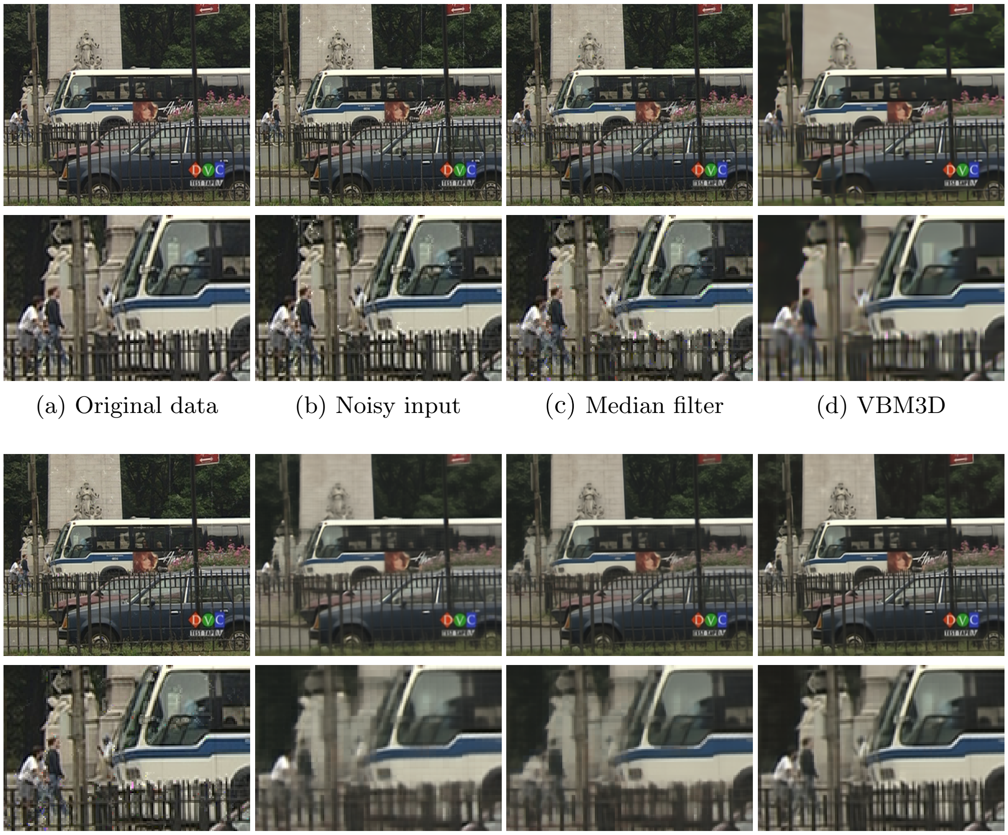
[14] Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Tran. Image Processing*, 19(5):1153 – 1165, 2010.

[15] A. Wong and J. Orchard. A nonlocal-means approach to exemplar-based inpainting. In *IEEE ICIP*, pages 2600–2603, 2008.

[16] D. Zhang and Z. Wang. Image information restoration based on long-range correlation. *IEEE Trans. Circuits Syst. Video Tech.*, 12(5):331–341, 2002.

[17] L. Zhang, S. Vaddadi, H. Jin, and S. K. Nayar. Multiple view image denoising. In *ICCV*, 2009.

[18] E. J. Candes and B. Recht. Exact matrix completion via convex optimization. *Found. of Comput. Math*, 9:717–772, 2009.



(a) Original data (b) Noisy input (c) Median filter (d) VBM3D

(e) LRC (f) PCA (g) MC (h) Algorithm 2

FIG. 3.4. *Video in-painting: the full images frames are shown in the first row with their magnified regions shown in the second row. (a) the un-corrupted data; (b) the one corrupted by scratches and hairs; (c)–(h): the result from the adaptive median filtering; the VBM3D method ([5]); the LRC method ([16]); the PCA method ([17]); the MC method ([8]); and Algorithm 2 respectively.*

[19] H. Hwang and R.A. Haddad. Adaptive median filters: new algorithms and results. *IEEE Trans. Image Processing*, 4(4):499–502, 1995.

[20] T. Chen and H. Wu. Adaptive impulse dectection using center-weighted median filters. *IEEE Signal Processing Letters*, 8(1):1–3, 2001.

[21] J.-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982.

[22] B. Liu and A. Zaccarin. New fast algorithm for the estimation of block motion vector. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(2), 1993.

[23] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 2010.

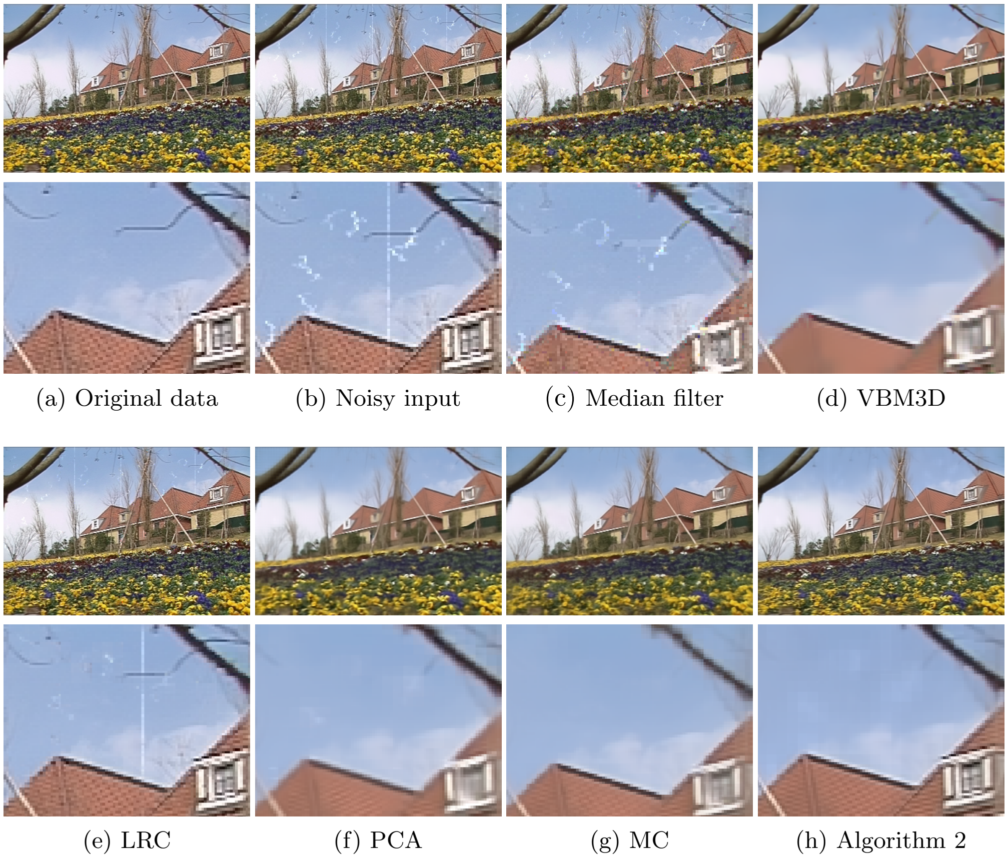
[24] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[25] Z. Shen, K.C. Toh, and S. Yun. An accelerated proximal gradient algorithm for frame based image restoration via the balanced approach. Technical report, National University of Singapore, 2009. preprint, 2009.

[26] K.C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific J. Optimization*, 6:615–640, 2010. In press, 2009.

[27] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Uilu-eng-09-2214, UIUC, Jul. 2009.

[28] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. Technical report, HKBU, 2009.



(a) Original data (b) Noisy input (c) Median filter (d) VBM3D

(e) LRC (f) PCA (g) MC (h) Algorithm 2

Fig. 3.5. *Video in-painting: the full images frames are shown in the first row with their magnified regions shown in the second row. (a) the un-corrupted data; (b) the one corrupted by scratches and hairs; (c)–(h): the result from the adaptive median filtering; the VBM3D method ([5]); the LRC method ([16]); the PCA method ([17]); the MC method ([8]); and Algorithm 2 respectively.*

[29] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Uilu-eng-09-2215, UIUC, Oct. 2009.

[30] Tao and Yuan. Recovering low-ran and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optimization*, 21(57), 2011.

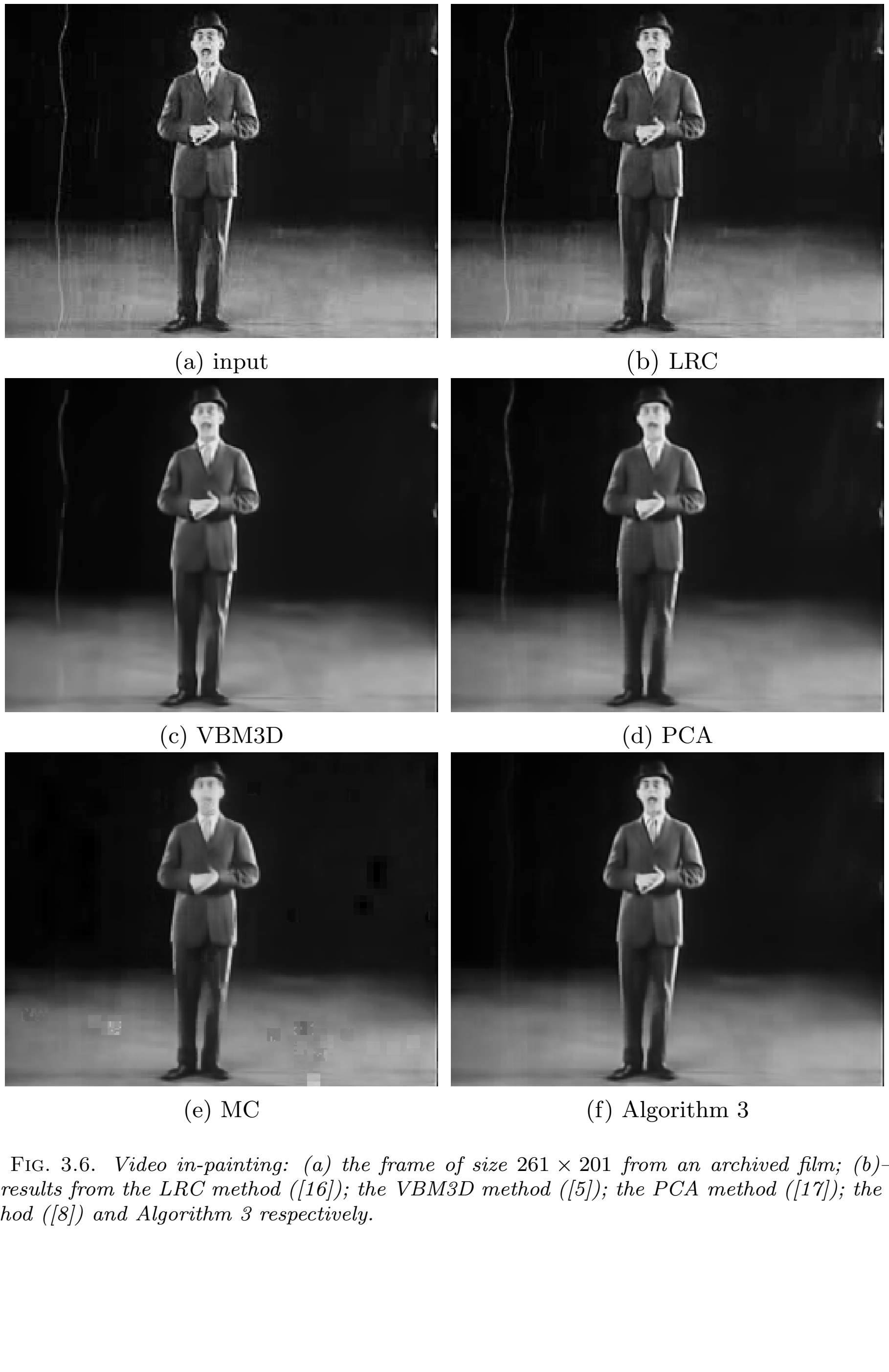
[31] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Trans. PAMI*, 16(3):267–276, 1994.

[32] A.C. Bovik. *Handbook of Image and Video Processing*. Academic Press Inc., San Diego, CA, 2000.

[33] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir. Detection and removal of line scratches in motion picture films. In *IEEE CVPR*, 1999.

[34] L. Joyeux, S. Boukir, B. Besserer, and O. Buisson. Reconstruction of degraded image sequences. application to film restoration. *Image Vis. Comput.*, (19):503–516, 2001.

[35] A. C. Kokaram. On missing data treatment for degraded video and film archives: a survey and a new bayesian approach. *IEEE Trans. Image Processing*, 13(3):397–415, 2004.



(a) input

(b) LRC

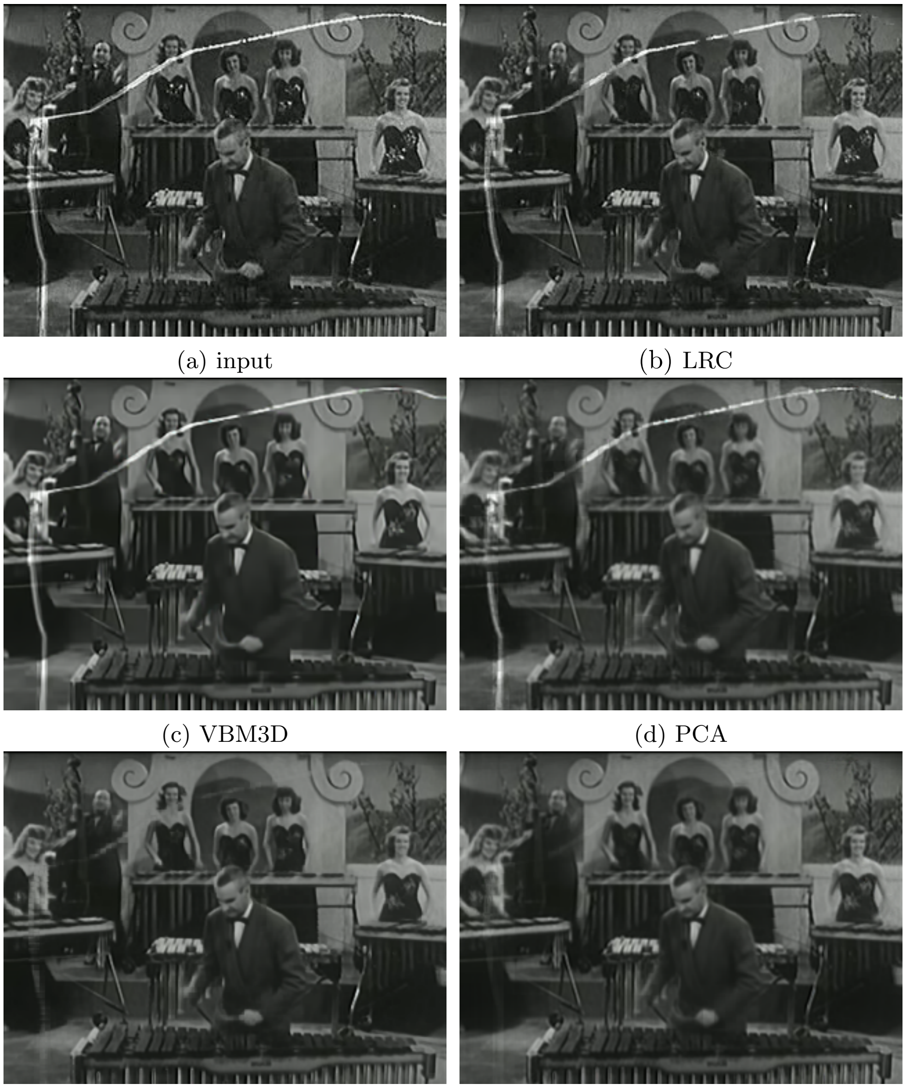
(c) VBM3D

(d) PCA

(e) MC

(f) Algorithm 3

FIG. 3.6. *Video in-painting: (a) the frame of size* 261 × 201 *from an archived film; (b)–(f): the results from the LRC method ([16]); the VBM3D method ([5]); the PCA method ([17]); the MC method ([8]) and Algorithm 3 respectively.*



(a) input

(b) LRC

(c) VBM3D

(d) PCA

(e) MC

(f) Algorithm 3

Fig. 3.7. *Video in-painting: (a) the frame of size* 320×240 *from an archived film; (b)–(f): the results from the LRC method ([16]); the VBM3D method ([5]); the PCA method ([17]); the MC method ([8]) and Algorithm 3 respectively.*