Single-patch low-rank prior for non-pointwise impulse noise removal

Ruixuan Wang Emanuele Trucco
School of Computing, University of Dundee, UK
{ruixuanwang, manueltrucco}@computing.dundee.ac.uk

Abstract

This paper introduces a 'low-rank prior' for small oriented noise-free image patches: considering an oriented patch as a matrix, a low-rank matrix approximation is enough to preserve the texture details in the properly oriented patch. Based on this prior, we propose a single-patch method within a generalized joint low-rank and sparse matrix recovery framework to simultaneously detect and remove non-pointwise random-valued impulse noise (e.g., very small blobs). A weighting matrix is incorporated in the framework to encode an initial estimate of the spatial noise distribution. An accelerated proximal gradient method is adapted to estimate the optimal noise-free image patches. Experiments show the effectiveness of our framework in removing non-pointwise random-valued impulse noise.

1. Introduction

This paper aims to remove random-valued impulse noise (RVIN) with varying sizes and irregular shapes (so called 'non-pointwise' RVIN, e.g., small particles suspended in the water; see Section 6.4). Based on the observation that almost any properly-oriented (defined later) small noise-free image patch (being a matrix) can be approximated by a low-rank patch with texture details well preserved, we propose a single-patch method to simultaneously detect and remove RVIN within a generalized joint low-rank and sparse matrix recovery framework. While the original matrix recovery framework has been recently used for image and video denoising [11], it requires multiple similar image patches, with each patch vectorized as a column in the matrix. Such a multi-patch method often limits the size of image patches to be relatively small (e.g., 8×8 pixels) in order to more likely find multiple similar patches within a single image and to collaboratively and effectively remove traditional (single-pixel) RVIN. Its performance degrades with non-pointwise RVIN, as shown in our experiments even with noisy regions as small as 3×3 pixels (Section 6.3). Instead, our single-patch method completely avoids searching for similar patches, and importantly, using larger-size patches (e.g., 40×40) in our method allows to effectively remove non-ponintwise RVIN.

1.1. Related work

We briefly discuss related work on impulse noise removal and low-rank matrix recovery; see [2] and [12] for recent, comprehensive reviews on denoising.

There are mainly two types of impulse noise: salt-and-pepper noise (black or white), and RVIN (any gray value). The median filter and its extensions are often effective to remove salt-and-pepper noise but can corrupt some textures. To reduce undesired corruption, various two-stage methods have been developed, first detecting the locations of noisy pixels, then recovering intensities only at noisy locations using certain filtering or variational methods, e.g., adaptive center weighted median filter (ACWMF) [4], rank-ordered absolute difference (ROAD) noise detector followed by a trilateral filtering [9], and a logarithmic version of the ROAD followed by edge-preserving regularization (EPR) for pixel restoration (ROLD-EPR) [7]. Both ROAD-trilateral and ROLD-EPR methods can preserve edges better than median-type methods like ACWMF as they consider local structures during pixel restoration. However, the accuracy of these two-stage methods depends crucially on the performance of the location stage. If noisy pixels cannot be detected correctly, e.g., when noise is structured rather than single-pixel, the overall noisy removal will be limited.

Given the excellent performance of non-local methods [2, 5], learned sparse models [8, 18], and the combination of both [6, 17] for random Gaussian noise, they were explored for impulse noise as well [20, 22]. Non-local methods use redundant visual information within an image (i.e., self-similarity) to group similar image patches together, followed by collaborative filtering [2, 5]. Sparse methods also use redundant information by assuming each patch can be well approximated by a linear combination of a small subset of patches ('words') within a large dictionary. Both types of methods can preserve texture details very well. However, patch size is limited (e.g., 8×8 pixels) as it may become difficult to find multiple similar larger-size patches within an image for non-local methods, and to represent a

1550-5499/13 $31.00 © 2013 Crown Copyright
DOI 10.1109/ICCV.2013.137

larger-size patch by a linear combination of other patches for sparse methods. Crucially, impulse noise often needs to be detected first to reduce the effect of noisy pixels on patch matching and dictionary learning. Similar to two-stage methods, the overall accuracy of impulse noise removal is largely limited by the performance of the initial impulse noise location.

A joint low-rank and sparse matrix recovery framework was recently used to detect and remove impulse noise simultaneously [11], remove background, and remove shadows and specularities from face images [3]. However, the method is limited by the usage of multiple similar patches and the small size of patches.

Unlike the above, this paper uses a different type of prior information, the 'low-rank prior', not for the whole image but for each image patch. It has been observed that low-rank textures exist in image regions having deterministic regular or periodic patterns [14, 21]; we found that almost any small (e.g., 10×10 to 40×40) image patch from images of natural or man-made objects or scenes (within our experiments, see Section 6), if rotated by a characteristic orientation defined later, has a low-rank approximation with texture details (including edges) well preserved (see Section 6.1).

2. Problem formulation

When a rectangular gray image patch $\mathbf{P}$ contains random Gaussian noise and RVIN, $\mathbf{P}$ may be decomposed as $\mathbf{P} = \mathbf{L}^* + \mathbf{S}^* + \mathbf{N}^*$, where $\mathbf{L}^*$ represents the unknown noise-free patch, $\mathbf{S}^*$ represents the unknown impulse noise (precisely, the difference between the noise and the corresponding noise-free pixel values), and $\mathbf{N}^*$ is a matrix of Gaussian noise [11].

$\mathbf{L}^*$ can be considered a low-rank matrix due to the low-rank prior for single patches (Section 6.1). Also, since the number of pixels corrupted by impulse noise is generally much smaller than the total number of pixels, $\mathbf{S}^*$ is a sparse matrix. As a result, the problem of image denoising can be formulated as an optimization problem [11], i.e., to minimize $E_1(\mathbf{L}, \mathbf{S})$ over the matrix variables $\mathbf{L}$ and $\mathbf{S}$,

$$E_1(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{P} - \mathbf{L} - \mathbf{S}\|_F^2 \;, \quad (1)$$

where $\|\cdot\|_*$ is the nuclear norm (i.e., sum of singular values) and considered as a convex relaxation of the rank measurement of the matrix; $\|\cdot\|_1$ is the sum of the absolute values of all matrix entries and considered as a convex relaxation of the sparsity measurement (i.e., number of non-zero entries) of a matrix; $\|\cdot\|_F$ is the Frobenius norm; and λ, μ are two regularization parameters.

While the above optimization framework has been used for image and video denoising [11], each image patch was considered as a column in a matrix. Such a multi-patch method is subject to the limitations discussed in Section 1.

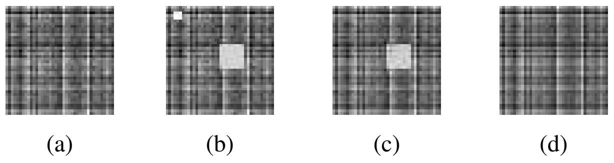


Figure 1: Illustrative example of low-rank matrix recovery. (a) A synthetic clean image patch of size 40×40 pixels with rank 20. (b) A synthetic noisy patch by adding a smaller (3×3 pixels) non-pointwise RVIN at the top-left corner and a larger (9×9 pixels) one around the center. (c) The larger noise has been largely remained by minimizing Equation (1). (d) Both are removed by minimizing Equation (2).

Now more specifically, using larger-size patches in the multi-patch method will generally make multiple patches less similar to each other and hence lead to over smoothing of the current patch (Figures 9 and 10). Compared to the multi-patch method, our method requires no search as it considers a single patch as the matrix $\mathbf{P}$ and the patch size can be larger (e.g., 41×41) without decreasing the efficiency of matrix recovery. More importantly, using larger-size patches allows us to remove non-pointwise RVIN.

The limited ability of the optimization framework (Equation 1) to remove non-pointwise impulsive noise is another key limit we address. The minimization of $E_1(\mathbf{L}, \mathbf{S})$ will generally lead to relatively small $\|\hat{\mathbf{L}}\|_* + \lambda\|\hat{\mathbf{S}}\|_1$ (and $\frac{1}{2\mu}\|\mathbf{P} - \hat{\mathbf{L}} - \hat{\mathbf{S}}\|_F^2$ as well) at the estimated optimal solution, $\hat{\mathbf{L}}$ and $\hat{\mathbf{S}}$. However, if very different (e.g., with much higher intensity value than the signal) non-pointwise impulse noise exists, the true solutions $\mathbf{S}^*$ and $\mathbf{L}^*$ often correspond to a much larger $\lambda\|\mathbf{S}^*\|_1$ (due to the set of higher impulse noise values) and modestly smaller $\|\mathbf{L}^*\|_*$ than those at the estimated solution $\hat{\mathbf{S}}$ and $\hat{\mathbf{L}}$. So the the minimum may not correspond to the true solution, i.e., $E_1(\mathbf{L}^*, \mathbf{S}^*) > E_1(\hat{\mathbf{L}}, \hat{\mathbf{S}})$, and the non-pointwise impulse noise will remain, to some extent, in the estimated optimal signal $\hat{\mathbf{L}}$ (Figure 1c).

Hence we propose a generalized version of the optimization framework to denoise an image patch effectively in the presence of non-pointwise (multi-pixel) RVIN:

$$E_2(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{W} \circ \mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{W} \circ (\mathbf{P} - \mathbf{L} - \mathbf{S})\|_F^2 \quad (2)$$

where $\mathbf{W}$ is a soft-weighting matrix with each entry value in $[0, 1]$, and $\circ$ denotes the Hadamard (i.e., entry-wise) product of two matrices. When every entry in $\mathbf{W}$ is set to 1, $E_2(\mathbf{L}, \mathbf{S})$ becomes $E_1(\mathbf{L}, \mathbf{S})$.

$\mathbf{W}$ can encode the initially estimated spatial distribution of impulse noise in the image patch. Initial estimates, obtained by any impulse noise detector, correspond to entries in $\mathbf{W}$ with values close to 0. Compared to the original optimization framework, Equation (1), the true solution, $\mathbf{S}^*$, will more likely correspond to a much smaller $\lambda \|\mathbf{W} \circ \mathbf{S}^*\|_1$, because (at least part of) the higher impulse noise values are counterbalanced by the corresponding smaller entry values in $\mathbf{W}$. From the regularization point of view, smaller entry values at initially estimated impulse noise locations in $\mathbf{W}$ will decrease the regularization effect of the second and the third terms in $E_2(\mathbf{L}, \mathbf{S})$, such that the entries at the corresponding locations in the matrix variable $\mathbf{L}$ can be searched in a larger feasible region in order to get a smaller $\|\mathbf{L}\|_*$. As a result, the ground-truth solution $\mathbf{L}^*$ will more likely correspond to the minimum of $E_2(\mathbf{L}, \mathbf{S})$. Therefore, $\mathbf{W}$ is expected to effectively help recover the corrupted signals (Figure 1d).

3. Optimization

The target function E_2 in Equation (2) can be minimized by accelerated proximal gradient (APG) [15, 19], which was recently developed to solve the original optimization problem, Equation (1). Noticing that $\mathbf{W}$ is a constant matrix, we can extend the original APG method to minimize $E_2(\mathbf{L}, \mathbf{S})$. More specifically, substituting $\mathbf{S} \leftarrow \mathbf{W} \circ \mathbf{S}$ and $\mathbf{P} \leftarrow \mathbf{W} \circ \mathbf{P}$, E_2 becomes

$$E_3(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{P} - \mathbf{W} \circ \mathbf{L} - \mathbf{S}\|_F^2 . \tag{3}$$

The only difference between E_3 and E_1 is the entry-wise weighting of $\mathbf{L}$ in the third term. Since such a difference does not change the conditions under which APG is applied, i.e., the cost function consists of a non-smooth convex function $g(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1$ and a smooth convex function $f(\mathbf{L}, \mathbf{S}) = \frac{1}{2\mu} \|\mathbf{P} - \mathbf{W} \circ \mathbf{L} - \mathbf{S}\|_F^2$ with its gradient $\nabla f(\mathbf{L}, \mathbf{S})$ Lipschitz continuous, APG to minimize E_1 can be directly extended to minimize E_3. For details see Algorithm 1, where $\mathcal{S}_\tau(\mathbf{Z})$ is the entry-wise shrinkage operator on the matrix $\mathbf{Z}$, i.e., $\mathcal{S}_\tau(z) = \mathtt{sgn}(z) \max(|z| - \tau, 0)$ for any element z in $\mathbf{Z}$ and $\mathtt{sgn}(z)$ is the sign of z. As in the original APG [15, 19], a continuation technique is applied in Algorithm 1 to reduce the number of necessary iterations by varying μ, i.e., starting from a large initial value μ_0 and then geometrically decreasing ($\rho\mu_k$) over iterations until it reaches the floor $\overline{\mu}$.

The main difference between Algorithm 1 and the original APG one is in Steps 2, 6, and 7, where $\mathbf{W}$ plays its role in the recovery of the low-rank matrix $\mathbf{L}$. Note that the output $\mathbf{S}$ is the weighted sparse matrix. The final $\mathbf{S}$ can be easily estimated from the difference between the input $\mathbf{P}$ and the output $\mathbf{L}$ (see [22])

4. Computing the weighting matrix W

The soft-weighting matrix $\mathbf{W}$ is a required input to the proposed method (see Algorithm 1). The basic procedure to generate $\mathbf{W}$ is as follows. First, the candidate impulse noise locations in an image patch are estimated by any of the methods suggested below to obtain a binary weighting matrix $\overline{\mathbf{W}}_0$, in which each entry is set to 1 at the initially estimated impulse noise pixels and 0 elsewhere. Then $\overline{\mathbf{W}}_0$ is convolved with a un-normalized 2D Gaussian operator $G(i,j) = \exp\{-\frac{1}{2\sigma_0^2}(i^2 + j^2)\}$, to generate a soft version $\overline{\mathbf{W}}$, where entries with higher value than 1 are set to 1. The final weighting matrix, $\mathbf{W}$, is set to $\mathbf{1} - \overline{\mathbf{W}}$, where $\mathbf{1}$ is a matrix with every entry equal to 1. Consequently, the entries of $\mathbf{W}$ at or near the initially estimated impulse noise locations will have smaller values than elsewhere.

Algorithm 1 APG method to minimize Equation (2)

```
Input: $\mathbf{P}, \mathbf{W}, \lambda$.
1: $\mathbf{L}_0 = \mathbf{L}_{-1} = \mathbf{0}; \mathbf{S}_0 = \mathbf{S}_{-1} = \mathbf{0}; t_0 = t_{-1} = 1; 0 < \overline{\mu} < \mu_0; 0 < \rho < 1$;
2: $\mathbf{P} \leftarrow \mathbf{W} \circ \mathbf{P}$;
3: while not converged do
4:     $\mathbf{Y}_k^L = \mathbf{L}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{L}_k - \mathbf{L}_{k-1})$;
5:     $\mathbf{Y}_k^S = \mathbf{S}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{S}_k - \mathbf{S}_{k-1})$;
6:     $\mathbf{G}_k^L = \mathbf{Y}_k^L - \frac{1}{2}\mathbf{W} \circ (\mathbf{W} \circ \mathbf{Y}_k^L + \mathbf{Y}_k^S - \mathbf{P})$;
7:     $\mathbf{G}_k^S = \mathbf{Y}_k^S - \frac{1}{2}(\mathbf{W} \circ \mathbf{Y}_k^L + \mathbf{Y}_k^S - \mathbf{P})$;
8:     $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = svd(\mathbf{G}_k^L), \mathbf{L}_{k+1} = \mathbf{U}\mathcal{S}_{\mu_k/2}(\mathbf{\Sigma})\mathbf{V}^{\mathrm{T}}$;
9:     $\mathbf{S}_{k+1} = \mathcal{S}_{\lambda\mu_k/2}(\mathbf{G}_k^S)$;
10:    $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}; \mu_{k+1} = \max(\rho\mu_k, \overline{\mu})$;
11:    $k \leftarrow k + 1$;
12: end while
Output: $\mathbf{L} \leftarrow \mathbf{L}_k, \mathbf{S} \leftarrow \mathbf{S}_k$.
```

In practice, the binary matrix $\overline{\mathbf{W}}_0$ can be generated by any existing impulse noise detectors (e.g., ROAD [9] or ROLD [7]), or even by our low-rank matrix recovery framework by setting $\mathbf{W} = \mathbf{1}$. This is because our framework (when $\mathbf{W} = \mathbf{1}$) also recovers the sparse matrix $\mathbf{S}$ representing the initial spatial distribution of impulse noise. Note that the denosing performance of the proposed method is robust to occasional errors in the estimated $\mathbf{W}$. If certain noise-free pixels occasionally have smaller weights in $\mathbf{W}$, such lower-weighted noise-free pixel values won't be changed too much in the final denoised low-rank patch because the rank of the patch is already small without changing these pixel values, while changing pixel values will probably increase the value of the second cost term in Equation (2).

5. Characteristic orientation for each patch

One potential issue in denoising methods is edge blurring and loss of sharpness. Patch orientation affects the result of our rank-based method. For example, even a patch with a simple pattern may have a high rank (Figure 2, patch in blue rectangle). In this case, the low-rank approximation of the patch will blur the sharp edge (Figure 2b). Instead, if we can find a low-rank patch (Figure 2, patch in yellow rectangle) by rotating around the target image point,

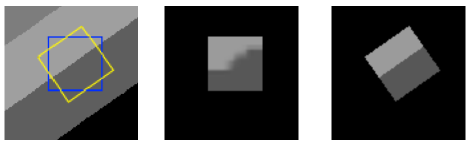

(a) (b) (c)

Figure 2: Effect of characteristic orientation on low-rank patch approximation. (a) A synthetic image. (b) A low-rank (rank 15) approximation of a 41×41 image patch around the image center. (c) The low-rank (rank 1) approximation of an oriented image patch around the same point.

the low-rank approximation of this patch will more likely preserve the sharpness of the edges (Figure 2c). Similar observations apply for patches with other texture patterns like corners, and experiments (Figure 8) show the importance of characteristic orientation in denoising.

Based on the assumption that the optimally oriented patch is low-rank, we expect that the difference between the oriented patch and its low-rank approximation will be minimum at the optimal ('characteristic') orientation. To compute the latter, let $\mathbf{P}(\theta)$ denote an oriented $m \times n$ image patch at a given image position, rotated anticlockwise by an orientation angle θ with respect to the image row direction, and $\tilde{\mathbf{P}}(\theta)$ the low-rank approximation (to a fixed quality level) of $\mathbf{P}(\theta)$. Then the characteristic orientation at the current image position can be estimated as

$$\hat{\theta} = \arg \min_{\theta \in [0,\pi]} d(\theta) = \arg \min_{\theta \in [0,\pi]} \frac{1}{mn} \left\| \mathbf{P}(\theta) - \tilde{\mathbf{P}}(\theta) \right\|^2 , \tag{4}$$

where $\| \cdot \|$ is a matrix norm (we use Frobenius), and θ is restricted in the range $[0, \pi]$ because any pair of θ and $\theta + \pi$ will lead to the same norm value.

In the above, 'quality level of the low-rank approximation' should be defined and a threshold introduced. However we found that a basic, efficiently computed rank-1 approximation of $\mathbf{P}(\theta)$ to represent $\tilde{\mathbf{P}}(\theta)$, i.e., every column in $\tilde{\mathbf{P}}(\theta)$ is the mean of all columns in $\mathbf{P}(\theta)$, leads to very good overall results especially in preserving edges. Such a simple, threshold-free approximation proved effective enough to find reliably the characteristic orientation for each image patch (see Section 6.2).

6. Experiments

Our APG algorithm (Algorithm 1) was implemented by modifying the public MATLAB source code for the original APG [15]. Similar to the parameter settings in [3], $\mu_0 = 0.99\|\mathbf{W} \circ \mathbf{P}\|_2$ and $\overline{\mu} = 10^{-9}\mu_0$. For an $m \times n$ image patch $\mathbf{P}$, $\lambda = 1/\sqrt{\max(m,n)}$. The maximum iteration number is set to 200. To find the characteristic orientation $\hat{\theta}$, a simple uniform sampling method was adopted with sampling interval $\frac{\pi}{36}$. σ_0 and the window size for the un-normalized Gaussian filter G were set to $\frac{1}{36}\min(m,n)$ and $\frac{1}{6}\min(m,n)$ respectively. When using the initially estimated sparse matrix $\mathbf{S}$ to generate $\overline{\mathbf{W}}_0$, the candidate noise locations are the entries where the absolute value is larger than a threshold, determined adaptively based on the expected sparsity level of the noise (i.e., number of noisy pixels over total pixel number). We use a sparsity level of 0.05.

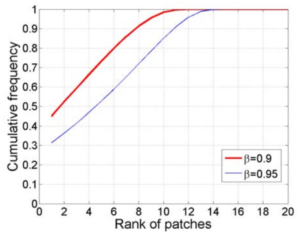
All the tests were performed using Matlab R2010a running on an Intel Core i7-2600K 3.40GHz PC with 8.0GB RAM. For an image with size 640×480 pixels, and patches with size 41×41 pixels and 50% overlapped by neighboring patches along both directions, totally it takes approximately 3 minutes to generate the denoised image. Since each image pixel is often covered by multiple patches, the final denoised value at each pixel in the image is averaged from the corresponding denoised pixels in these multiple patches.

6.1. Low-rank prior in single patches

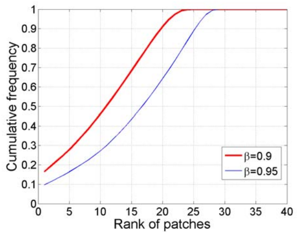
We illustrate the experimental foundation of the low-rank prior for small noise-free image patches. In practice, due to noise, an image patch as a matrix is seldom low-rank. However, if the assumption of low-rank prior is true, the column-wise signal variation in an image patch should be mostly preserved in a much lower-dimensional space, and the low-rank approximation should preserve meaningful textural details. Such predictions are confirmed empirically from the following tests with the public datasets Caltech256 [10] and SceneCategory15 [13]. Around $450,000$ oriented (i.e., rotated to their characteristic orientations) patches with sizes 21×21 and 41×41 pixels were generated from each dataset by uniform sampling in each image.

The first test explores statistically the order of the low-rank approximation needed to preserve, at a given level β, the column-wise signal variation in each $m \times n$ image patch. Every patch is first decomposed by SVD and the minimum number $\hat{l}$ of singular values necessary to preserve the predefined level of signal variation is determined by $\hat{l} = \arg\min_{J \in [1, \min(m,n)]} \{\beta < \sum_{j=1}^{J} \sigma_j / \sum_{j=1}^{\min(m,n)} \sigma_j\}$, where σ_j is the j-th largest singular value. Different patches may have different rank values $\hat{l}$. A rank histogram can be easily generated recording the frequency of patches with a particular rank value. The cumulative rank histogram in Figure (3a) (thinner blue line) shows that when preserving 95% of the column-wise signal variations, about 90% image patches have low-rank approximations with rank equal or smaller than 11. When $\beta = 0.9$, more than 98% image patches have low-rank approximations with rank equal or smaller than 10. Similar results were obtained for various sizes in both datasets (Figure 3b). This shows that most oriented patches can be approximated by their low-rank versions which keep most signal variations.

The second test shows that low-rank patch approxima-

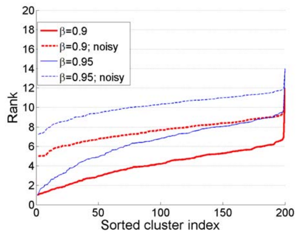


(a)

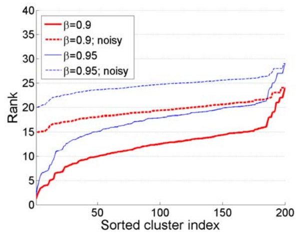


(b)

Figure 3: Low-rank prior in image patches with size (a) 21×21 (from Caltech256) and (b) 41×41 pixels (from SceneCategory15).



(a)

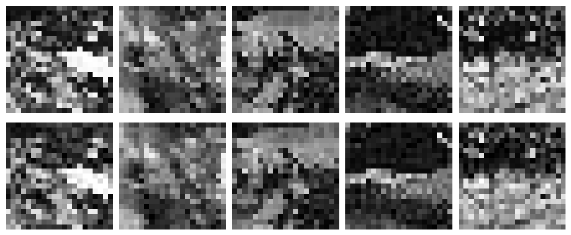


(b)

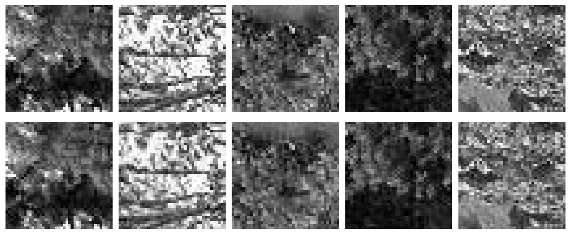
Figure 4: Average rank value of each cluster for image patches with size (a) 21×21 (from Caltech256) and (b) 41×41 pixels (from SceneCategory15).

tions can preserve textural details. A fast k-means method [1] was applied to cluster the patches of a fixed size into 200 clusters using the datasets above. The average $\hat{l}$ over all patches was computed within each cluster. Figure 4 (solid curves) shows the sorted average rank values for all the clusters. Consistent with the first test, when $\beta = 0.95$, most clusters have average rank values less than 10 for 21×21 patches and less than 20 for 41×41 patches. The small subset of clusters with larger average rank values often correspond to the patches with more complex visual appearance or patterns. Within each such cluster, the highest-rank ($\hat{l}$) image patch was chosen to represent the most complex visual pattern. Figure 5a lists such image patches and the corresponding low-rank approximations. It can be observed that, even for the patches with most complex texture patterns, the textural details have all been preserved in the low-rank approximations. Similar observations have been found for the 41×41 image patches (Figure 5b). While such observations are seemingly trivial, it is actually not true for (at least some) un-oriented patches (see Figure 2b).

In addition, when adding RVIN to the image patches by corrupting 5% pixels in each patch, Figure 4 (dashed curves) also shows that the average rank values increased. It suggests that not only the noise-free image patches are low-rank, but also the noisy patches have higher ranks at a predefined level β. The proposed low-rank matrix recovery framework just makes use of this observation for removing noise from image patches.



(a)



(b)

Figure 5: Image patches for the 5 clusters with highest average rank values, with patch size (a) 21×21 pixels (top: original; bottom: rank-10 approximation) and (b) 41×41 pixels (top: original; bottom: rank-20 approximation).

6.2. Determination of characteristic orientation

The first test here checks whether the determination of characteristic orientation is invariant to changes in patch sizes. As an example, we use an image patch (Figure 6a top) from a noisy and low-contrast underwater hydrocolonoscopy image (Figure 8a). Figure (6) shows that for a large range of patch sizes (e.g., 21×21 to 41×41), the estimated characteristic orientations are almost the same (see the minimum of each curve in Figure 6b). This invariance to patch sizes is especially beneficial to denoising images with textures at different scales.

The second test checks whether the orientation determination is robust to noisy patches. Given an image patch (Figure 7 top left), Figure (7) shows that the estimate of the characteristic orientation is robust to RVIN noise, even if 30% of image pixels have been damaged by RVIN.

The third test compares the proposed method with the well-known 'SIFT-orientation' method [16], where the dominant gradient direction was determined based on the weighted histogram of gradient orientations. The image was smoothed twice by a Gaussian with standard devision 1.6 and window size $6 * 1.6$ pixels, and then gradient with 4 pixels spacing along both dimensions was computed at every location. Different Gaussian and spacing parameters were tried with results similar to Figure (8b). Compared to the results by the proposed method (Figure 8c), the char-

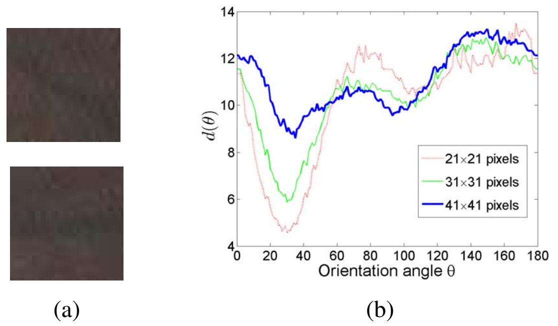


Figure 6: Determination of characteristic orientation. (a) An original low-contrast 41×41 image patch (top) and the oriented version (bottom) (better view on monitor). (b) $d(\theta)$ over all possible orientation angles with three patch sizes.

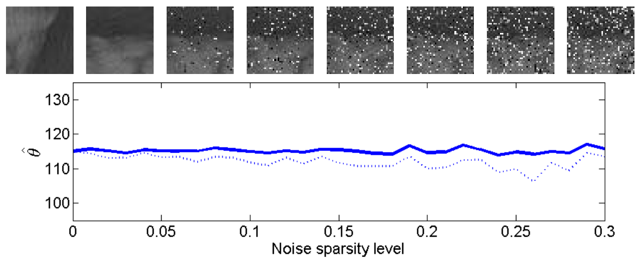


Figure 7: Robustness of characteristic orientation determination. Top row (left to right): original image patch, oriented patches with noise sparsity level at 0.00, 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30. Bottom: estimated characteristic orientation with varying noise sparsity levels, with mean (solid curve) and standard deviation (dotted curve) values from 10 runs.

acteristic orientations by the SIFT-orientation method are often not precisely orthogonal to the dominant edge in each image patch, probably due to the noisy environment. In the denoised result, based on the proposed matrix recovery framework, certain edges have been blurred when using SIFT-orientation method (Figure 8e). In comparison, the sharpness of edges has been preserved by the proposed method (Figure 8f). While sampling error is introduced to the oriented patches due to rotation of the original patches, the image quality loss due to sampling error appears to be much smaller compared to image enhancement due to noise removal from the oriented patches. This is supported by Figures 8d-8f, which shows that both orientation methods perform better than without using any orientation determination (Figure 8d).

6.3. Removal of impulse noise

To evaluate our method quantitatively, RVIN with different sizes (e.g., 1×1 to 4×4 pixels) at a particular sparsity level (i.e., 0.1) was added to the noise-free images to generate the noisy images. Although only two well-known images ('Barbara' and 'Lena') were used to demonstrate the performance due to limited space, similar results have been obtained for other widely-used standard images (e.g., Baboon, Finger, etc). The denoising performance was measured by standard peak signal-to-noise ratio (PSNR).

Four denoising methods were chosen for comparison: the median filtering as the baseline method, the ROLD-EPR method [7], the multi-patch low-rank matrix recovery method (MPLR) [11], and the proposed method without applying the weighting matrix $\mathbf{W}$ (henceforth 'Proposed $\backslash\mathbf{W}$'). For median filtering, 20 iterations were run with window size 3×3 pixels. The PSNR was computed after each iteration and the highest PSNR value reported. Similarly for ROLD-EPR, the maximum iteration was set 20, the window size was 5×5 and all the other parameters were set as suggested in [7]. The highest PSNR over all iterations was reported. For MPLR, all the parameters were set as suggested in [11], except that four different patch sizes (i.e., $n \times n, n \in \{4, 8, 16, 32\}$) were tried and 10 additional similar patches were searched (to generate a $n^2 \times 11$ matrix) across the whole image when denoising each image patch. The highest PSNR over the different patch sizes was reported. For our method, the patch size was fixed to 31×31, the highest PSNR was reported over different $\lambda = s/\sqrt{31}$, where $s \in \{1.0, 1.3, 1.6, 2.0\}$. For each method, 10 runs were performed, with PSNR standard deviation around 0.1.

Table 1 (last two rows) shows that the proposed method consistently performs comparable or better than 'Proposed $\backslash\mathbf{W}$' (26.45 and 26.40 for 'Barbara' 3×3 RVIN are not significantly different). The substantial improvement in PSNR for large-size (i.e., 4×4) RVIN is due to the effect of the weighting matrix $\mathbf{W}$. Table 1 also shows that the proposed method performs better than the other methods with non-pointwise RVIN (i.e., 3×3 and 4×4). This is probably due to the limited ability to detect and remove large-size RVIN by the ROLD-EPR method, and the oversmoothing by median filtering and MPLR, as demonstrated in Figure 9c. For the image with more textured regions (i.e., 'Barbara'), the proposed method also performs better even when RVIN is smaller (i.e., 1×1 and 2×2). Figure 9a and 9a show that noticeable noise still remains after denoising by ROLD-EPR, and the regular patterns have been blurred by MPLR. Consistent with PSNR assessment, the proposed method gave the best visual quality. For the image with less textured regions (i.e., 'Lena'), ROLD-EPR performs best for RVIN of size 2×2. However, careful inspection found that about 20% RVIN were still noticed in the denoised image by ROLD-EPR. In comparison, only 8% RVIN were noticed in the denoised image by the proposed method. This indicates that the higher PSNR by ROLD-EPR is largely from keeping the non-corrupted pixels from changed rather than removing more RVIN. In addition, the

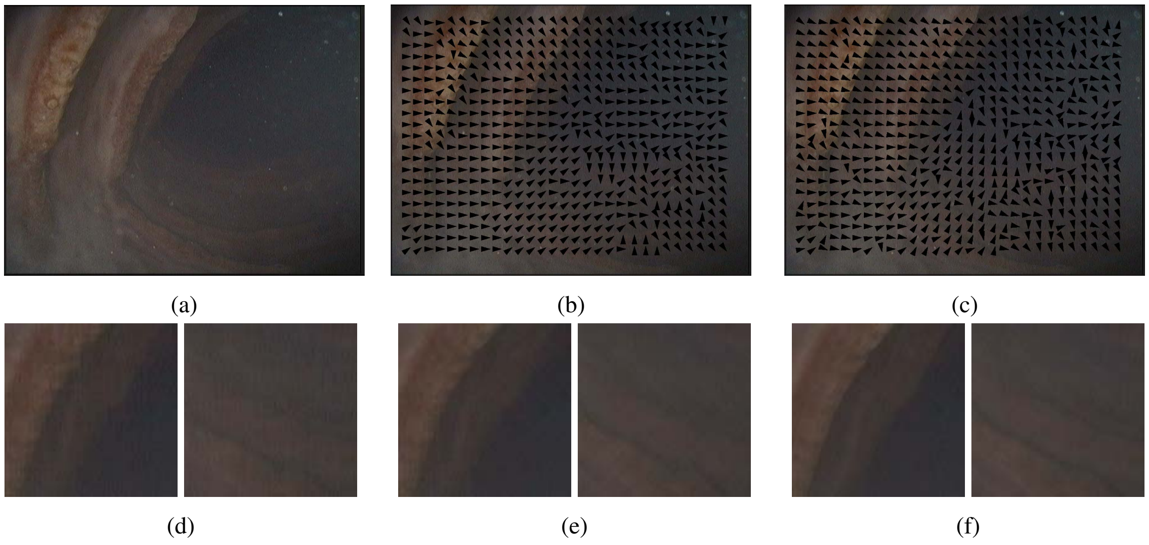


Figure 8: Characteristic orientation and its effect on denoising. (a) The original image. The orientation (by arrow) determined by (b) the SIFT-orientation method and (c) the proposed method for 41×41 patches. Two cropped denoised image regions (d) without orientation determination, (e) using the SIFT-orientation method, and (f) using the proposed method.

Table 1: PSNR from different methods for different sizes of RVIN.

	Barbara				Lena			
	1×1	2×2	3×3	4×4	1×1	2×2	3×3	4×4
Noisy image	19.02	19.09	19.00	18.93	19.43	19.52	19.56	19.30
Median	24.90	23.90	23.28	18.94	33.64	30.78	28.65	20.37
ROLD-EPR	29.48	26.80	18.99	18.90	35.89	**33.58**	19.63	19.34
MPLR	28.60	25.95	24.06	23.61	**37.18**	31.34	27.28	26.58
Proposed $\setminus \mathbf{W}$	29.69	28.13	**26.45**	23.99	33.32	30.99	28.65	25.15
Proposed	**30.69**	**28.43**	26.40	**25.34**	34.44	31.12	**28.90**	**27.30**

superior performance by MPLR when RVIN size is 1×1 simply confirms the previous-study result when combining the non-local idea with low-rank matrix recovery [11].

6.4. Particle removal in hydrocolonoscopy images

The proposed denoising method is also capable of removing particles in hydrocolonoscopy images, where the particles with various sizes (maximumly 15×15 pixels) were suspended in water, creating non-pointwise RVIN. About 1000 hydrocolonoscopy images were extracted from a video captured by a colonoscope moving within a reversed-engineered phantom of a human colon segment immersed in turbid water in a tank. Figure 10 displays an exemplar hydrocolonoscopy image and the denoised result with the MPLR and the proposed method. The patch size is 41×41 for the proposed method and 16×16 for the MPLR method in order to remove large-size particles. MPLR over-smoothed the image with larger particles partially remaining (Figure 10c and 10d). The 'Proposed $\setminus \mathbf{W}$' can preserve sharpness but cannot remove large-size particles effectively (Figure 10e and 10f). The proposed method can remove

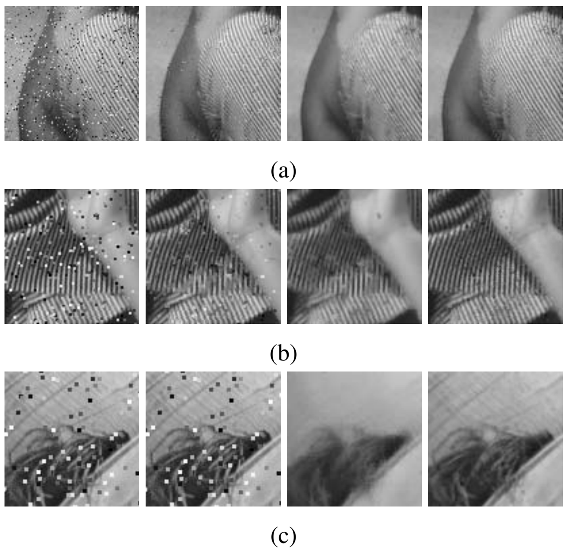


Figure 9: Part of denoised images from different methods with noise size (a) 1×1, (b) 2×2, and (c) 3×3 pixels. From left to right in column: noisy patch, denoised result by ROLD-EPR, MPLR, and the proposed method.

the particles effectively while preserving the sharpness of the edges (Figure 10g and 10h). Similar results have been obtained for other hydrocolonoscopy images.

7. Conclusions

This paper has introduced a low-rank prior for small oriented (rotated by a characteristic orientation angle) noise-

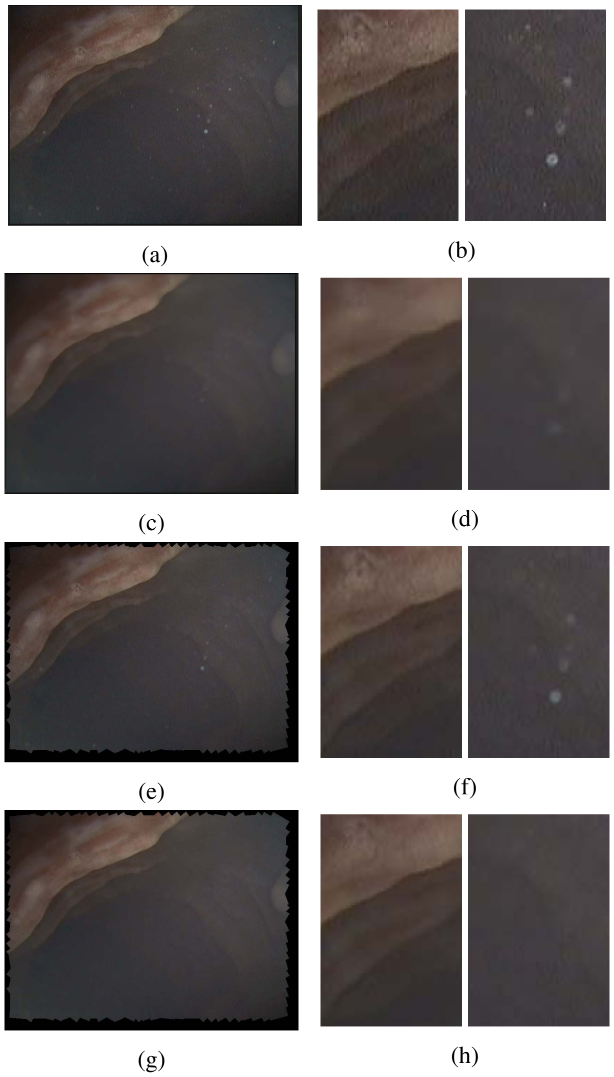


Figure 10: Removal of particles suspended in a hydrocolonoscopy image. (a) The original noisy image. (b) Denoised image by MPLR with patch size 16×16 pixels. (c) Denoised image by the proposed method without $\mathbf{W}$ (dark boundary regions not denoised). (d) Denoised by the proposed method. (e-h) Two image patches cropped from each of the images in (a-d).

free image patches. The low-rank prior suggests that a single patch can be effectively denoised within a low-rank matrix recovery framework. Without resorting to other similar patches, the single-patch method can effectively remove non-pointwise RVIN within a generalized low-rank matrix recovery framework, and encode the initial estimation of noise locations effectively. Experimental results show the better performance of the proposed approach over several methods, especially for non-pointwise RVIN. Removing random Gaussian noise and video denoising will be explored as future work.

Acknowledgment: This work is funded by 2011-2016 EU FP7 ERC project CODIR: colonic disease investigation by robotic hydrocolonoscopy, collaborative between the Universities of Dundee (PI Prof Sir A Cuschieri) and Leeds (PI Prof A Neville). Thanks to Dr A Alazmani and Mr M Pakleppa for hydrocolonoscopy video capture, and Prof P Lin and CVIP members for comments and discussions.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA*, 2007. 5

[2] A. Buades, B. Coll, and J. Morel. Image denoising methods. a new nonlocal principle. *SIAM Review*, 52(1):113–147, 2010. 1

[3] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *JACM*, 58(3), 2011. 2, 4

[4] T. Chen and H. Wu. Adaptive impulse dectection using center-weighted median filters. *SPL*, 8(1):1–3, 2001. 1

[5] K. Dabov, V. Katkovnik, R. Foi, K. Egiazarian, and S. Member. Image denoising by sparse 3D transform-domain collaborative filtering. *TIP*, 16(8):2080–2095, 2007. 1

[6] W. Dong, X. Li, L. Zhang, and G. Shi. Sparsity-based image denoising via dictionary learning and structure clustering. In *CVPR*, 2011. 1

[7] Y. Dong, R. Chan, and S. Xu. A detection statistic for random-valued impulse noise. *TIP*, 16(4):1112–1120, 2007. 1, 3, 6

[8] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 15(12):3736–3745, 2006. 1

[9] R. Garnett, T. Huegerich, C. Chui, and W. He. A universal noise removal algorithm with an impulse detector. *TIP*, 14(11):1747–1754, 2006. 1, 3

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset, 2007. CNS-TR-2007-001. 4

[11] H. Ji, S. Huang, Z. Shen, and Y. Xu. Robust video restoration by joint sparse and low rank matrix approximation. *SIAM J. Imaging Sci.*, 4(4):1122–1142, 2011. 1, 2, 6, 7

[12] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola. From local kernel to nonlocal multiple-model image denoising. *IJCV*, 86(1):1–32, 2010. 1

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 4

[14] X. Liang, X. Ren, Z. Zhang, and Y. Ma. Repairing sparse low-rank texture. In *ECCV*, 2012. 2

[15] 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. Technical Report UILU-ENG-09-2214. 3, 4

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009. 1

[18] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *TIP*, 17(1):53–69, 2008. 1

[19] K. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific J. Optimization*, 6(3):615–640, 2010. 3

[20] Y. Xiao, T. Zeng, J. Yu, and M. Ng. Restoration of images corrupted by mixed gaussian-impulse noise via $l_1 - l_0$ minimization. *PR*, 44(8):1708–1720, 2011. 1

[21] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform invariant low-rank textures. *IJCV*, 99(1):1–24, 2012. 2

[22] Y. Zhou, Z. Ye, and Y. Xiao. A restoration algorithm for images contaminated by mixed gaussian plus random-valued impulse noise. *J. Vis. Commun. Image R.*, 99:283–294, 2013. 1, 3