Robust Video Denoising *

Rishi Agarwal
rishiagarwal@cse.iitb.ac.in

June 20, 2020

1 Problem Statement

Most existing video denoising algorithms assume a single statistical model of image noise, e.g. additive Gaussian white noise, which often is violated in practice. The problem is to denoise a video having multiple noise sources. For such videos no assumption of noise model should be made.
Let $F = \{f_k\}_{k=1}^{K}$ be the image sequence with K frames. each image f_k is a sum of its underlying clean image g_k and the noise n_k:

$$f_k = g_k + n_k \tag{1.1}$$

The goal of video denoising is to recover $G = \{g_k\}_{k=1}^{M}$ $(M \leq K)$ by removing n_k from f_k. This new approach for video denoising was presented by Ji, Liu, Shen, and Xu (2010)

2 Main Algorithms

We use a two-stage video denoising approach. To exploit the temporal redundancy in a video, we take a patch-based approach to jointly remove image noise n_k for all image frames.
In the first stage, we apply adaptive median filtering on the frames and use these filtered frames to find 'm' matching patches for a reference patch 'p'. Using these vectorized patches we generate a matrix 'M'. This matrix has some elements which are reliable and others which are not reliable. We retain all the reliable elements and discard others. Lets call the obtained matrix 'Q'.
In the second stage, we exploit the fact that a matrix of similar patches should have low rank and apply low rank matrix recovery algorithm on the matrix 'Q'. Now we have the denoised pixel values for the patches forming the matrix 'Q'.
We repeat this process for all the patches. Now for each pixel we take average of all the values obtained for it. This gives us the denoised frames.

*This a report for a course project at IITB

2.1 Ranked-order based Adaptive Median Filter

The RAMF algorithm is based on a test for the presence of an impulse at the center pixel followed by a test for the detection of a residual impulse in the median filter output. For detecting the presence of an impulse we simply check if the pixel value is equal to the extreme values (i.e. 0 or 255 in our case).
If no impulse is present at the center pixel, we do nothing with it. If impulse is detected, we find the median filter output with window size 'w'. If the output value is not an impulse, we replace the center pixel with median filter output. However, if the median filter output is an impulse we increase the window size 'w' to '$w + 1$'.
This simple modification has much better results as compared to the naive median filtering.
This algorithm was presented by Hwang and Haddad (1995)

2.2 Fixed Point Iterative Algorithm

This algorithm falls under the "Forward Backward Splitting" class of algorithms.
We want to solve for:

$$min_X ||X||_*$$

$$s.t.\ ||X|_\Omega - Q|_\Omega||_F^2 \leq \#(\Omega)\hat{\sigma}^2 \tag{2.2.1}$$

where,
Ω is the set of indices for which the elements of matrix 'Q' are reliable,
$\hat{\sigma}$ is the estimate of standard deviation of noise, which is obtained by calculating the average of the variances of all elements $\epsilon\ \Omega$ on each row.
Instead of solving (2.2.1) directly, we solve its Lagrangian version:

$$min_X\ \frac{1}{2}||X|_\Omega - Q|_\Omega||_F^2 + \mu||X||_* \tag{2.2.2}$$

Let $f(X) = \frac{1}{2}||X|_\Omega - Q|_\Omega||_F^2\ ,\ g(X) = \mu||X||_*$.
We solve this iteratively. In each iteration we first perform a simple forward gradient descent step on f. This step begins at iterate X^k, and then moves in the direction of the (negative) gradient of f, which is the direction of steepest descent. The second step is called the proximal step, or backward gradient descent step which solves:
$min_X\ \frac{1}{2}||X - \hat{X}^{k+1}||_F^2 + \gamma g(X)$

Algorithm 1: Fixed point iterative algorithm

Result: X^k
$X^0 := 0$;
while *not converged* **do**
  $\hat{X}^{k+1} = X^k - \gamma(X^k|_\Omega - Q|_\Omega)$;
  $X^{k+1} = prox_{\gamma g}(\hat{X}^{k+1}) = D_{\gamma\mu}(\hat{X}^{k+1})$
end

D is the shrinkage operator as defined:

$$D_\tau(X) = U\Sigma_\tau V^T \tag{2.2.3}$$

where,
$X = U\Sigma V^T$
$\Sigma_\tau = max(\Sigma - \tau I, 0)$
$1 \leq \gamma \leq 2$ and μ are pre-defined parameters.
$\mu = (\sqrt{n_1} + \sqrt{n_2})\sqrt{p}\hat{\sigma}$, where $n_1 \times n_2$ is the size of matrix 'Q', $p = \frac{\#(\Omega)}{n_1 n_2}$
We state the proof for correctness of the algorithm in a later section.

3 Datasets

In this section, we evaluate the performance of the proposed method on several video samples corrupted by different types of mixed noise. All the video data used in the experiments can be downloaded from this website. We are using 64×64 size image frames for each video shown in figures: 3, 4, 6. For figure 5, we used 128×128 size image frames. By default, we use $K = 3, 5$ image frames, set patch size to be 8×8 pixels. Set the range of image intensity to [0, 255]. For each reference patch, 5 most similar patches are used in each image frame based on l_1 norm distance function. We are adding Gaussian noise (σ), Poisson noise (κ) and impulsive noise (s) to each frame.

4 Results

In this section we present our experimental results on the datasets mentioned above.
The following table shows the advantage of adaptive median filtering over normal median filtering.

Window size	Normal Median Filtering	Adaptive Median Filtering
3	17.285	24.6692
4	19.8752	24.3247
5	21.6647	24.8468

Table 1: Comparison of PSNR values of images obtained from normal median filtering to those obtained from adaptive median filtering. Noise level ($\sigma = 10$, $\kappa = 40$, $s = 40\%$).

The following figures show that adaptive median filtering yields more visually pleasant denoised results as compared to median filtering with fixed window size.

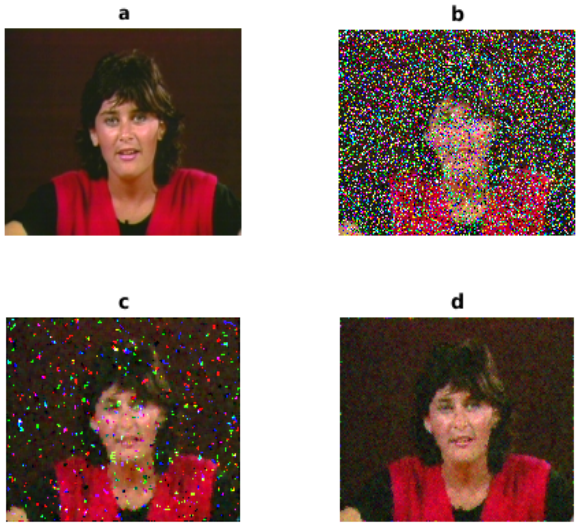

Figure 1: (a) Actual image without noise, (b) Noisy image, (c) Image after normal median filtering, (d) Image after adaptive median filtering

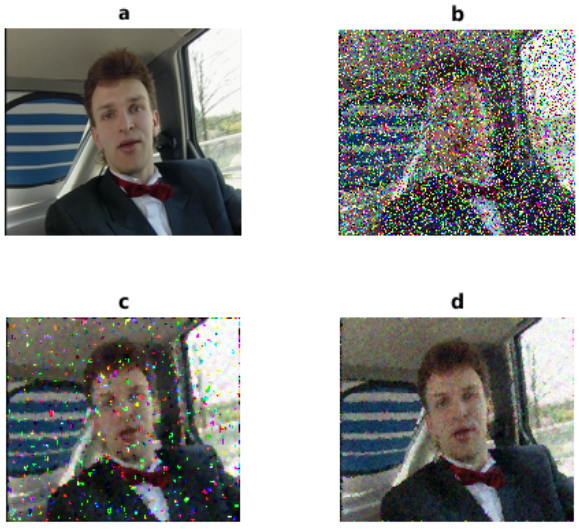

Figure 2: (a) Actual image without noise, (b) Noisy image, (c) Image after normal median filtering, (d) Image after adaptive median filtering

The tables 1, 2 and 3 were generated for the dataset named 'miss_am' on the website. We present the denoised frames obtained from our main algorithm for some of the datasets below.

s/κ	**10**	**20**	**30**	**40**
20%	30.6731	30.5734	30.586	30.4264
30%	30.2121	30.2585	30.2619	30.0594
40%	29.2625	29.561	28.9507	29.3339

Table 2: the PSNR values of the denoised images with respect to different noise levels of Poisson noise and impulsive noise. Gaussian noise level is fixed at $\sigma = 10$. This table shows robustness of algorithm to increase in κ and s.

σ	**s**	κ	PSNR
10	40%	40	29.3339
15	30%	30	29.6194
20	20%	30	29.2700
20	40%	40	27.1099
30	20%	10	27.6083
30	40%	40	26.4351

Table 3: the PSNR values of the denoised images with respect to different noise levels. This table is intended to show the variation of PSNR with σ.

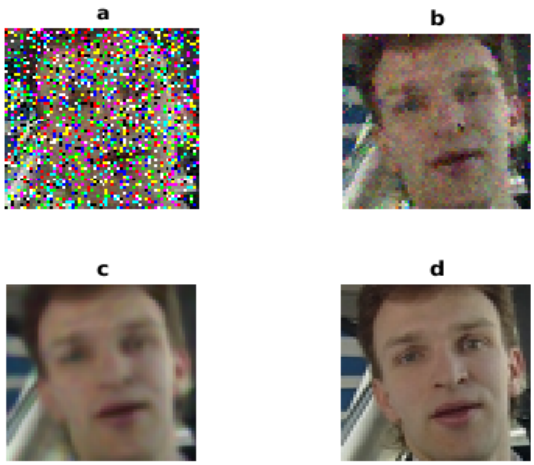

Figure 3: (a) Noisy image, (b) Image after median filtering, (c) Image after low rank matrix recovery, (d) Actual image without noise, Noise level ($\kappa = 40$, $s = 40\%$, $\sigma = 10$)

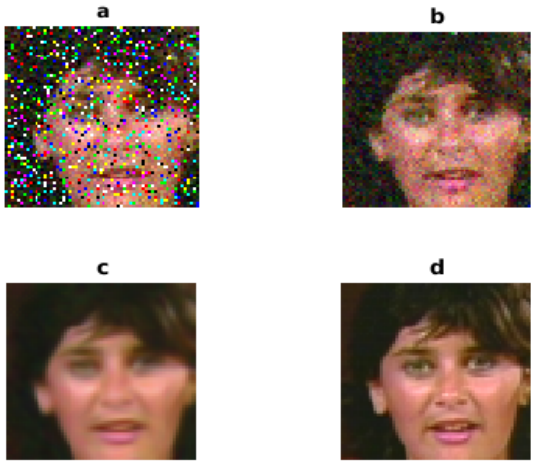

Figure 4: (a) Noisy image, (b) Image after median filtering, (c) Image after low rank matrix recovery, (d) Actual image without noise, Noise level ($\kappa = 30$, $s = 20\%$, $\sigma = 20$)

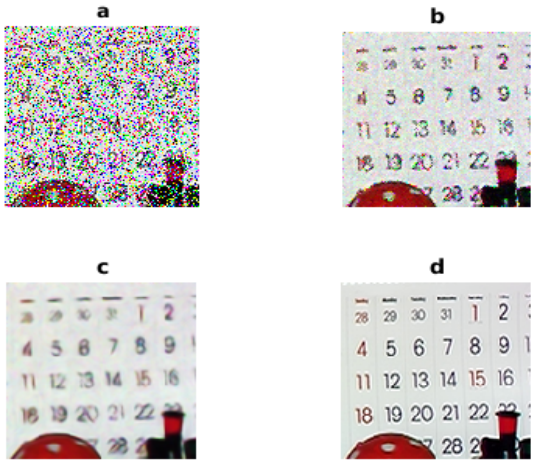

Figure 5: (a) Noisy image, (b) Image after median filtering, (c) Image after low rank matrix recovery, (d) Actual image without noise, Noise level ($\kappa = 40$, $s = 40\%$, $\sigma = 10$)

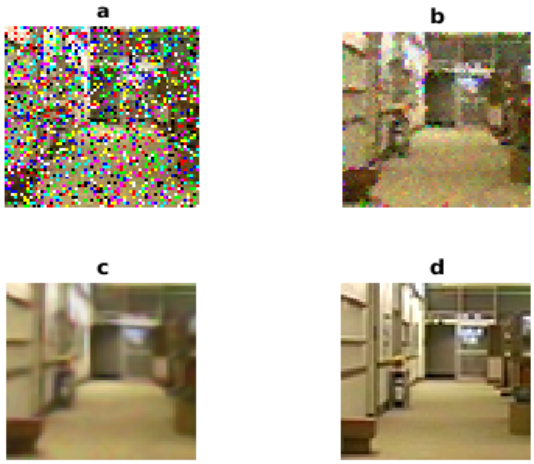

Figure 6: (a) Noisy image, (b) Image after median filtering, (c) Image after low rank matrix recovery, (d) Actual image without noise, Noise level ($\kappa = 40$, $s = 40\%$, $\sigma = 10$)

5 Discussion and Conclusions

By formulating the video denoising problem to a low-rank matrix completion problem, our algorithm does not assume any specific statistical properties of image noise. The algorithm requires us to choose an important parameter- the step size (γ), which affects the performance and reliability of the algorithm. We are running our algorithm patch-wise, which makes it very slow for large frame sizes. Choice of K is also very important, as large K leads to blurring and loss of sharpness and small K may not carry sufficient information.

6 Proof of correctness of FPI algorithm

In this section, two proofs have been presented. The first one is an easier proof, but its results are weak. The second one is a more involved proof, which has stronger results than the earlier one. We only present the outline for the second proof as it is higly non-trivial.

6.1 Proof of convergence of algorithm - 1 (Easier version)

Some notations:

$$x^k = prox_{\gamma f_1}(x^{k-1} - \gamma \nabla f_2(x^{k-1})) \tag{6.1.1}$$

$$f(x) = f_1(x) + f_2(x)$$

where, both are convex, f_2 differentiable, and ∇f_2 L-Lipschitz continuous. Let $\{x^k\}$ be defined by Equation (6.1.1) and let $0 < \gamma \leq 1/L$.
At the k^{th} step of our method we obtain x^k by minimizing the function

$$G_k(x) = f(x) + g_k(x)$$

It is something very similar to the approach we had studied in ISTA. The auxillary function g_k is chosen appropriately, such that it satisfies the inequality:

$$0 \leq g_k(x) \leq G_{k-1}(x) - G_{k-1}(x^{k-1}) \quad \forall x \tag{6.1.2}$$

The inequality (6.1.2) implies that $g_k(x^{k-1}) = 0$. We also assume that inf $f(x) = b > -\infty$.

Lemma 6.1 *The sequence $\{f(x^k)\}$ is non-increasing and the sequence $\{g_k(x^k)\}$ converges to zero.*

Since x^{k+1} minimizes $G_{k+1}(x)$ and $g_{k+1}(x^k) = 0$, we have:

$$G_{k+1}(x^{k+1}) \leq G_{k+1}(x_k)$$

$$f(x^{k+1}) + g_{k+1}(x^{k+1}) \leq f(x^k) + g_{k+1}(x^k)$$

$$f(x^k) - f(x^{k+1}) \geq g_{k+1}(x^{k+1}) \geq 0$$

$\{f(x^k)\}$ is decreasing to a finite limit, since it is bounded by b. Thus, the sequence $\{g_k(x^k)\}$ converges to 0.

Lemma 6.2 *The sequence $\{f(x^k)\}$ converges to b.*

Suppose $\exists \delta > 0$ s.t. $f(x^k) \geq b + 2\delta \; \forall$ k.
$\exists z$ s.t. $f(x^k) \geq f(z) + \delta \; \forall$ k as inf $f(x) = b$
Using (6.1.2),

$$f(z) + g_k(z) - f(x^k) - g_k(x^k) \geq g_{k+1}(x)$$

$$g_k(z) - g_{k+1}(z) \geq f(x^k) + g_k(x^k) - f(z) \geq f(x^k) - f(z) \geq \delta$$

This is not possible as successive differences of a non increasing sequence $\{g_k(z)\}$ of non negative terms must converge to 0. Therefore, the sequence $\{f(x^k)\}$ converges to b.
Now we make a guess for g_k,

$$G_k(x) = f(x) + \frac{1}{2\gamma}||x - x^{k-1}||_2^2 - D_{f_2}(x, x^{k-1})$$

where,

$$D_{f_2}(x, x^{k-1}) = f_2(x) - f_2(x^{k-1}) - \langle \nabla f_2(x^{k-1}), x - x^{k-1} \rangle$$

Since f_2 is convex, $D_{f_2}(x, y) \geq 0 \; \forall$ x,y.

$$g_k(x) = \frac{1}{2\gamma}||x - x^{k-1}||_2^2 - D_{f_2}(x, x^{k-1})$$

Let,

$$h(x) = \frac{1}{2\gamma}||x||_2^2 - f_2(x)$$

Now, $g_k(x) = D_h(x, x^{k-1})$. $g_k(x) \geq 0$ whenever h is convex.
As ∇f_2 is L-lipschitz continuous and $0 < \gamma \leq 1/L$, $\forall$ x, y:

$$\gamma||\nabla f_2(x) - \nabla f_2(y)|| \leq ||x - y||$$

$$\gamma||\nabla f_2(x) - \nabla f_2(y)|| \times ||x - y|| \leq ||x - y||^2$$

By Cauchy Shwartz inequality,

$$\gamma\langle \nabla f_2(x) - \nabla f_2(y), x - y \rangle \leq \gamma||\nabla f_2(x) - \nabla f_2(y)|| \times ||x - y|| \leq ||x - y||^2$$

$$\frac{1}{\gamma}\langle (x - \gamma\nabla f_2(x)) - (y - \gamma\nabla f_2(y)), x - y \rangle \geq 0$$

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq 0$$

Thus, h is convex. So, $g_k(x) \geq 0$.

Lemma 6.3 *x^k given by Equation* (6.1.1) *minimizes G_k.*

We know that $x = prox_f(z)$ minimizes $\frac{1}{2}||x - z||^2 + f(x)$.
Therefore,

$$0 \ \epsilon \ \partial(\gamma f_1)(x^k) + x^k - (x^{k-1} - \gamma \nabla f_2(x^{k-1}))$$

$$(x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k \ \epsilon \ \partial(\gamma f_1)(x^k) \tag{6.1.3}$$

Since x^k minimizes G_k if and only if,

$$0 \ \epsilon \ \nabla f_2(x^k) + \frac{1}{\gamma}(x^k - x^{k-1}) - \nabla f_2(x^k) + \nabla f_2(x^{k-1}) + \partial f_1(x^k)$$

$$(x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k \ \epsilon \ \partial(\gamma f_1)(x^k)$$

which is true by (6.1.3).
Thus, x^k indeed minimizes G_k.

Theorem 6.4 *Sequence $\{x^k\}$ converges to minimizer of $f(x)$ wherever minimizers exist.*

$$G_k(x) - G_k(x^k) = f_1(x) - f_1(x^k) + \frac{1}{2\gamma}||x - x^{k-1}||_2^2 - \frac{1}{2\gamma}||x^k - x^{k-1}||_2^2 + \langle \nabla f_2(x^{k-1}), x - x^k \rangle$$

Using $||x - x^k||_2^2 - 2\langle x^{k-1} - x^k, x - x^k \rangle = ||x - x^{k-1}||_2^2 - ||x^k - x^{k-1}||_2^2$, we get:

$$G_k(x) - G_k(x^k) = f_1(x) - f_1(x^k) + \frac{1}{2\gamma}||x - x^k||_2^2 - \frac{1}{\gamma}\langle x^{k-1} - x^k, x - x^k \rangle + \langle \nabla f_2(x^{k-1}), x - x^k \rangle$$

$$G_k(x) - G_k(x^k) = f_1(x) - f_1(x^k) + \frac{1}{2\gamma}||x - x^k||_2^2 - \frac{1}{\gamma}\langle x^{k-1} - x^k - \gamma \nabla f_2(x^{k-1}), x - x^k \rangle \tag{6.1.4}$$

Since, $(x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k \ \epsilon \ \partial(\gamma f_1)(x^k)$, by definition of sub-differential:

$$f_1(x) - f_1(x^k) - \frac{1}{\gamma}\langle x^{k-1} - x^k - \gamma \nabla f_2(x^{k-1}), x - x^k \rangle \geq 0$$

Thus,

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}||x - x^k||_2^2 \geq g_{k+1}(x) \tag{6.1.5}$$

Let $\hat{x}$ minimize $f(x)$, then:

$$G_k(\hat{x}) - G_k(x^k) = f(\hat{x}) + g_k(\hat{x}) - f(x^k) - g(x^k)$$

Using (6.1.2),

$$G_k(\hat{x}) - G_k(x^k) \geq f(\hat{x}) + (G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1})) - f(x^k) - g(x^k)$$

As $f(x^k) - f(\hat{x}) \geq 0$ and $g_k(x^k) \geq 0$,

$$(G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1})) - (G_k(\hat{x}) - G_k(x^k)) \geq f(x^k) - f(\hat{x}) + g_k(x^k) \geq 0$$

Thus, sequence $\{G_k(\hat{x}) - G_k(x^k)\}$ is non-negative and decreasing. The sequences $\{g_k(x^k)\}$ and $\{f(x^k) - f(\hat{x})\}$ converge to 0, as the successive differences of a decreasing non negative sequence should converge to 0.

Using (6.1.5), it follows that the sequence $\{x^k\}$ is bounded. Thus, by Bolzano-Weierstrass Theorem, a subsequence of $\{x^k\}$ converges to some x^* s.t. $f(x^*) = f(\hat{x})$
Replacing $\hat{x}$ with x^*, we find that $\{G_k(x^*) - G_k(x^k)\}$ is decreasing, and by equation (6.1.4) a subsequence, and therefore, the entire sequence converges to 0.
Thus, we conclude that using (6.1.5) that $\{||x^* - x^k||_2^2\}$ converges to 0. Thus, $\{x^k\}$ converges to x^*.
This proof was presented by Byrne (2014). Now lets use this theorem to prove correctness of our algorithm (1).
Let 'M' be a matrix and 'm' its vectorized form for any letter now onwards.
Note that to prove the theorem we used vectors, but our algorithm has matrices. However, we can handle this by vectorizing the matrices.
For our problem, $f_1(x) = \mu||X||_*$, $f_2(x) = \frac{1}{2}||x|_{\Omega'} - q|_{\Omega'}||_2^2$, where 'x' is vectorized form of 'X' and Ω' is the new set of indices.
∇f_2 is 1-lipschitz continuous operator, i.e.

$$||x|_{\Omega'} - y|_{\Omega'}|| = ||\nabla f_2(x) - \nabla f_2(y)|| \leq ||x - y||$$

f_1 is convex. The vectorized form of the matrix given by shrinkage operator (2.2.3) gives x^{k+1} which solves $min_x \frac{1}{2}||x - \hat{x^{k+1}}||_2^2 + \gamma f_1(x)$. Thus, the shrinkage operator is equivalent to (6.1.1).
Therefore, the theorem proves correctness of our algorithm for $0 < \gamma \leq 1$.
The second proof is required to prove the correctness of the algorithm for $0 < \gamma < 2$.

6.2 Proof of convergence of algorithm - 2 (Harder version)

$$x^k = prox_{\gamma f_1}(x^{k-1} - \gamma \nabla f_2(x^{k-1}))$$
$$x^k = \mathcal{A}x^{k-1} \tag{6.2.1}$$

When we select γ in the interval (0, 2/L), the operator $\mathcal{A}$ becomes averaged, since it is now the product of a fne (firmly non expansive) operator and an av (averaged) operator. Convergence of the sequence $\{x^k\}$ to a fixed point of $\mathcal{A}$, whenever $\mathcal{A}$ has fixed points, then follows from the Krasnosel'skii-Mann (KM) Theorem.
The details for this proof have been presented by Combettes and Wajs (2005).

References

Byrne, C. L. (2014, 1). An Elementary Proof of Convergence for the Forward-Backward Splitting Algorithm. *Journal of nonlinear and convex analysis*, *15(4)*.

Combettes, H., & Wajs, V. (2005). Signal recovery by proximal forwardbackward splitting. *Multiscale Modeling and Simulation*, *4(4)*, pp. 1168-1200. doi: 10.1137/050626090

Hwang, H., & Haddad, R. A. (1995, 4). Adaptive Median Filters: New Algorithms and Results. *IEEE Transactions on Image Processing*. doi: 10.1109/83.370679

Ji, H., Liu, C., Shen, Z., & Xu, Z. (2010, 6). Robust video denoising using Low rank matrix completion. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2010.5539849