ADVANCED IMAGE PROCESSING

PROJECT REPORT

ROBUST VIDEO DENOISING USING LOW RANK MATRIX COMPLETION

TEAM MEMBERS:	ROLL NO:
G SRAVANI	203070004
SACHIN PATIL	203079001

INTRODUCTION

The majority of available video de-noising algorithms rely on a single statistical model of image noise, such as additive Gaussian white noise, which is not always true. We implement a patch-based video de-noising algorithm based on [1]. The challenge of removing mixed noise is formulated as a low-rank matrix completion problem by grouping similar patches in both the spatial and temporal domains. This technique does not rely on strong assumptions about the statistical features of noise.

To simulate real life scenarios, the video is corrupted with Poisson noise, Gaussian noise and Impulse noise.

ALGORITHM

1. **Main Algorithm:**
   We first perform adaptive median filtering on each frame in order to find an initial subset of reliable pixels. Then, the image is divided into 8X8 overlapping patches, found with stride = 4 in both directions. For each 8X8 patch K (=5 in our implementation) similar patches are found across every frame using L1 distance as the similarity metric. These patches are further filtered to obtain the final subset of reliable pixels. Then the stack of matched patches are re-arranged to form a matrix. Such a matrix tends to be a noisy version of a low-rank matrix with several missing elements. This low rank matrix is then recovered by using the fixed point algorithm.

   The algorithm is run for each of the R,G and B components of each frame separately and then the de-noised components are then used to find the final colour image.

2. **Adaptive Median Filtering:**

This consists of two tests to find out whether a particular pixel is noisy or not.

Test I : Determine if the median found within a window lies within minimum and maximum values of that window. If yes, go to test B. Else, increase window size until the test is passed or window size reaches maximum allowed value. If window size has reached maximum, output the median value for that window.

Test 2: Determine if the pixel value lies within minimum and maximum values of that window. If yes, output the pixel value itself. Else, output is the median value within that window.

3. **Fixed Point Algorithm:**

1. Set $Q^{(0)} := 0$.
2. Iterating on k till $\|Q^{(k)} - Q^{(k-1)}\|_F \leq \epsilon$,

$$\begin{cases} R^{(k)} = Q^{(k)} - \tau \mathcal{P}_\Omega(Q^{(k)} - P), \\ Q^{k+1} = D_{\tau\mu}(R^{(k)}), \end{cases}$$

where μ and $1 \leq \tau \leq 2$ are pre-defined parameters, D is the shrinkage operator defined in (4) and $\mathcal{P}_\Omega$ is the projection operator of Ω defined by

$$\mathcal{P}_\Omega(Q)(i,j) = \begin{cases} Q(i,j), & \text{if } (i,j) \in \Omega; \\ 0, & \text{otherwise.} \end{cases}$$

3. Output $Q := Q^{(k)}$.

Here, $D_\tau(X) = U\Sigma_\tau V^T$, where $\Sigma_\tau = diag(\max(\sigma_i - \tau, 0))$ and $\mu = (\sqrt{n_1} + \sqrt{n_2})\sqrt{p}\hat{\sigma}$.
$n_1 :=$ total no. of elements in a patch; $n_2 :=$ total no. of selected patches for a particular reference patch.

4. **Minimization Problem:**

Rank minimization of Q is done by minimizing its nuclear norm using the fixed point algorithm:

$$\min_Q \left|\left|Q\right|\right|_*$$
$$\text{s.t.} \left(\; ||Q|_\Omega - P|_\Omega \; || \; \right)_F{}^2 \leq \#(\Omega)\hat{\sigma}^2$$

DATASET

We used two videos for experimentation, viz., *carphone_qcif.y4m* and *akiyo_cif.y4m.*

EXPERIMENTS

We conducted an experiment to determine the robustness of the algorithm. We fixed the Gaussian noise level at $\sigma = 10$ and varied the levels of Poisson noise (κ) and Impulse noise (s) as follows:

s/κ	5	10	20	30
10%	23.647	21.879	20.071	18.920
20%	22.745	21.367	19.744	18.263
30%	21.325	20.439	18.491	17.627

OBSERVATIONS

1. As s increases (and κ is kept fixed), there is very little change in the PSNR values which shows the high robustness towards impulse noise.
2. However, PSNR values drop when Poisson noise level is increased and impulse noise is fixed. Though, overall it handles all the three noises reasonably.

VISUAL OUTPUTS

1. *Carphone_qcif* (frame 17)

Original | Corrupted | Adaptive Median Filter Output | Final output

2. *Akiyo_cif* (frame 17)

Original | Corrupted | Adaptive Median Filter Output | Final output