Fast dejittering for digital video images using local non-smooth and non-convex functionals

Mila Nikolova

CMLA, ENS Cachan, CNRS, UniverSud
61 Av. President Wilson, F-94230 Cachan, FRANCE

http://www.cmla.ens-cachan.fr/~nikolova
nikolova@cmla.ens-cachan.fr

Imaging with Modulated/Incomplete Data (SFB Workshop), July 2010, Graz Austria

Bayesian TV
MAE=11.7, PSNR=22

Bake & Shake
MAE=7.4, PSNR=23

Proposed ≡ Original
MAE=0, PSNR=∞
Jitter occurs in digital video when the synchronization signal is corrupted, in fax documents

Original image $f \in \mathbb{R}^{r \times c}$, jittered image $g \in \mathbb{R}^{r \times c}$ ($f_i$ is a row of $f$, $g_i$ is a row of $g$):

$$1 \leq j \leq c, \ 1 \leq i \leq r, \ g_i(j) = \begin{cases} f_i(j + d_i) & \text{if } 1 \leq j + d_i \leq c \\ 0 & \text{otherwise} \end{cases} \quad d_i \in \mathbb{Z}, \ |d_i| \leq M.$$  \ \ (1)

**Intrinsic dejittering = restore the image frame from the jittered data** (Kokaram 1997)

**Main existing methods:**

- A. Kokaram et al. (1997 and 1998) : 2D AR model + drift compensation
- J. Shen (2004) : Fully bayesian method with TV prior (B-TV)

**Our goal:** For each jittered row $g_i$ we wish to estimate its displacement $\hat{d}_i$ based on the previously restored rows $\hat{f}_{i-1}, \hat{f}_{i-2}, \ldots$

$\Rightarrow$ We need a good prior on the gray-value of the columns of natural images
1. Choice of a criterion

The gray-value of the columns of natural images can be seen as pieces of $2^{\text{nd}}$ or $3^{\text{rd}}$ order polynomials which is hard to claim for their jittered versions.

For stability, the lowest degree differences that fit real-world images is better

⇒ Consider differences of the form \[ |g_i(j - d_i) - 2\hat{f}_{i-1}(j) + \hat{f}_{i-2}(j)| \]

Each row of $g$ has at one of its extremes at most $N$ (overestimate $\geq M$) null pixels

⇒ The $N$ columns at the extremities of $g$ are globally meaningless.
⇒ They must be excluded from our criterion.
⇒ The other columns $N + 1, \cdots, c - N$ bear sound information on the true image.
⇒ A criterion that uses only columns $N + 1, \cdots, c - N$
Estimated row displacement:

\[
\hat{d}_i = \arg \min \left\{ \mathcal{J}(d_i) : d_i \in \{-N, \cdots, 0, \cdots, N\} \right\}
\]

\[
\mathcal{J}(d_i) = \frac{1}{n - m + 1} \sum_{j=m}^{n} |g_i(j - d_i) - 2\hat{f}_{i-1}(j) + \hat{f}_{i-2}(j)|^\alpha, \quad \alpha \in \{0.5, 1\}
\]

\[
m = N + 1 + \max\{d_i, \hat{d}_{i-1}, \hat{d}_{i-2}\}
\]
\[
n = c - N + \min\{d_i, \hat{d}_{i-1}, \hat{d}_{i-2}\}
\]

Constants \(m, n\) guarantee that all indexes within the sum belong to \(\{N + 1, \cdots, c - N\}\)

Normalization \((n - m + 1)^{-1}\) is needed because for each shift \(d_i\), the sum contains a different number of terms while the solution \(\hat{d}\) is the minimizer over a set of \(\mathcal{J}(d_i)\)

Restored row:

\[
\hat{f}_i(j) = g_i(j - \hat{d}_i) \quad \text{if } 1 \leq j \leq c \text{ and } 1 \leq j - \hat{d}_i \leq c
\]
1st order looks for constant vertical pieces. 3rd order is too "loose" to discriminate between a column of a natural image and its slightly wrong displacements.
Interpretation of $\mathcal{J}$

- $\mathcal{J}$ finds a $\hat{d}_i$ such that $\hat{f}_i(j + \hat{d}_i) \approx 2\hat{f}_{i-1}(j) - \hat{f}_{i-2}(j)$ for a maximum number of $j$ at current row $i$. Constraint is stronger for $\alpha < 1$
- The contribution to $\mathcal{J}$ of $|\hat{f}_i(j + \hat{d}_i) - 2\hat{f}_{i-1}(j) + \hat{f}_{i-2}(j)|^\alpha \gg 0$ (a breakpoint) decreases as far as $\alpha \leq 1$ decreases—a vertical edge at $\hat{f}_i(j + \hat{d}_i)$ can be recovered.

$\Rightarrow \alpha \in (0, 1)$. For stability $|.|^\alpha$—increasing enough: we consider $\alpha \in \left[\frac{1}{2}, 1\right]$

- $i \in \{173, 298, 419, 478\}$
- $i \in \{23, 123, 273, 477\}$
- $i \in \{173, 298, 419, 478\}$
- $i \in \{23, 123, 273, 477\}$

$X$-axis: $d_i \in \{-M, \ldots, M\}$ for $M = 7$.

$Y$-axis: $\mathcal{J}(d_i) = \frac{1}{n - m + 1} \sum_{j=m}^{n} |f_i(j + d_i) - 2f_{i-1}(j) + f_{i-2}(j)|$.

$f$ is the original image and the true displacement of row $i$ is naturally $\hat{d}_i = 0$. 
2. Specific error measures

- The percentage of displaced rows in \( \hat{f} \) w.r.t. the original: 
  \[ e_0(\hat{d}, d) \overset{\text{def}}{=} \frac{100}{r} \|d - \hat{d}\|_0 \% \]

- The maximum horizontal error: 
  \[ e_\infty(\hat{d}, d) \overset{\text{def}}{=} \frac{100}{c} \|d - \hat{d}\|_\infty \% \]

- The changes in \( d - \hat{d} \): 
  \[ e_0^\Delta(\hat{d}, d) \overset{\text{def}}{=} \frac{100}{r-1} \# \{(d_i - d_i) - (\hat{d}_{i+1} - d_{i+1}) \neq 0, \ 1 \leq i \leq r-1 \} \% \]

The proposed method leads to essentially piecewise constant \( d - \hat{d} \) whose largest part = 0

Example: Consider that \( d - \hat{d} \) is composed of \( L \) pieces

\[
\{1, \cdots, i_2 - 1\}, \ \{i_2, \cdots, i_3 - 1\}, \ \cdots, \ \{i_L, \cdots, r\}
\]

where \( d_i = \hat{d}_i, i_2 \leq i \leq i_3 - 1 \) (the largest constant piece). Then

\[
e_0(\hat{d}, d) = i_2 - 1 + r - i_3 + 1 \quad \text{(can be high for any } L \geq 2)\]

\[
e_0^\Delta(\hat{d}, d) = L \times \frac{100}{r-1} \%
\]

By the latter, \( L \) pairs of consecutive rows are misplaced relatively to each other.

Let the maximal displacement between two consecutive rows is \( K \) pixels. Then

\[
e_\infty(\hat{d}, d) = K \times \frac{100}{c} \%
\]

For a \( 512 \times 512 \) image, i.e. \( r = c = 512 \), errors like \( L = 4 \) and \( K = 2 \) remain visually indistinguishable from the original. Their measures read \( e_0^\Delta = 0.78\% \) and \( e_\infty = 0.39\% \).
Remark: If both $e_\infty$ and $e_0^\Delta$ are small (e.g. $e_\infty \leq 0.4\%$ and $e_0^\Delta \leq 0.8\%$), we are guaranteed that dejittering is nearly perfect, independently of any other error measure.

E.g. for a $512 \times 512$ image—no more than 4 rows have a horizontal error up to 2 pixels. Such an error is invisible to the naked eye.

If one of these measures is higher, nothing can be claimed on the quality of $\hat{f}$.

E.g., if the image is planar (or constant) on a horizontal slice, large errors $e_\infty$ and $e_0^\Delta$ remain invisible.

Bayesian TV (B-TV)  
MAE=13.36, PSNR=20.82  

Bake & Shake (B&S)  
MAE=12.5, PSNR=20.27  

Proposed $\mathcal{J}$, $\alpha \in \{0.5, 1\}$  
MAE=0.16, PSNR=42.87  

$e_\infty=0.39\%$, $e_0^\Delta=0.25\%$
3. Main algorithm

1. \( g = [ g^L : \bar{g} : g^R ] \) where \( g^L \in \mathbb{R}^{r \times N} \), \( \bar{g} \in \mathbb{R}^{r \times (c-2N)} \) and \( g^R \in \mathbb{R}^{r \times N} \)

2. for each \( i \), optimal displacement \( \hat{d}_i \) is computed based on \( \bar{g}_i \) by minimizing \( J \)

3. then \( f_i(j + N) = g_i(j - \hat{d}_i), 1 \leq j \leq c \)

4. at step \( r \), \( \tilde{f} \in \mathbb{R}^{r \times (c+2N)} \) and \( \hat{f} \) is an \( r \times c \) submatrix of \( \tilde{f} \).

Note: \( \tilde{f}_i \in \mathbb{R}^{1 \times (c+2N)} \)

Computation times: For a 512 \( \times \) 512 image and \( N = 7 \), the solution is got 0.62 second for \( \alpha = 1 \) and in 1 second for \( \alpha = 0.5 \). (Matlab 7.2, PC with a Pentium 4 CPU 2.8GHz and 1GB RAM, Windows XP Professional service pack 2)

4. Large-scale experiment

1000 experiments with Lena, Peppers, Barbara, Boat for 2 different types of independent jitter.

- The means of \( e_0^\Delta \) and \( e_\infty \) are small.
- \( \hat{d} - d \) is either null or piecewise constant with values near 0.
- The mean and the variance of MAE are small.
- In almost all cases, \( \alpha = 0.5 \) is better than \( \alpha = 1 \).
Jitter Jittered image (512×512) Main Algorithm, $\alpha = 0.5$

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>4.16</td>
</tr>
<tr>
<td>PSNR</td>
<td>25.53</td>
</tr>
<tr>
<td>$e_1$</td>
<td>0.52</td>
</tr>
<tr>
<td>$e_\infty$</td>
<td>0.39%</td>
</tr>
<tr>
<td>$e_0^\Delta$</td>
<td>0.59%</td>
</tr>
</tbody>
</table>

Error $\hat{f} - f$ (error crosses the nose) Zoom dejittered Zoom original
Jittered Image $M=10$

Original

Our: $\alpha=0.5$, $\text{MAE}=1.35$, $\text{PSNR}=31.51$

Our: Error $f - \hat{f}$
5. Color images

The jitter is the same for all color channels (R,G,B)

In step 2 of Main Algorithm (sec. 3) $g_i$ is replaced by $|\bar{g}_i^R| + |\bar{g}_i^G| + |\bar{g}_i^B|$ (gray value)

The row shift $\hat{d}$ is estimated for the gray value image. Then $\hat{d}$ is inserted in the color image

Original

$\alpha = 1$

MAE = 0.2

PSNR = 40.34

$\alpha = 0.5$

$e_{\infty} = 0.24\%$

$e_0 = 0.37\%$
Jittered, $M=20$  \hspace{1cm} \alpha=0.5, \ e_\infty=0.35\%, \ e_0^{\Delta}=0.28$  \hspace{1cm} \text{Original 707}\times579
\[ \alpha = 1 \]
\[ \text{MAE} = 1.45 \]
\[ \text{PSNR} = 33.82 \]

Original

Errors mainly in flat strips
6. Noisy jittered images

For weak noise (≥ 20 – 30 dB) just use Main Algorithm

![20dB SNR+Jitter M=6](image1) ![Noisy, Dejittered, α = 1](image2) ![Denoising: hard shrinkage](image3) ![Original (256×256)](image4)

**Strong noise**

Main idea

For $i = 1, \cdots, r$

- Denoise row $\tilde{g}_i$ using fast shrinkage estimator to ensure the 2nd order assumption
- Estimate $\hat{d}_i$ using Main Algorithm

Insert $\hat{d}$ into jittered noisy data to get a noisy dejittered image

Use a denoising method to restore this noisy dejittered image
Changes in Main Algorithm:

• changes in step 2
  – if RGB image—transform \( \overline{g}_i \) into gray-value (as in sec. 5)
  – replace \( \overline{g}_i \) by \( \gamma_i = W^* y_i^T \) where
    for \( W : \mathbb{R}^{1 \times c} \to \mathbb{R}^{1 \times c} \) — 1D wavelet transform and \( W^* \) its inverse,
      \[
y_i = W \overline{g}_i \in \mathbb{R}^{1 \times c}
      \]
      and for a (small) \( T > 0 \), \( y_i^T (j) = \begin{cases} 0 & \text{if } |y_i(j)| \leq T \\ y_i(j) & \text{otherwise} \end{cases} \quad 1 \leq j \leq c.
  – replace \( \mathcal{J} \) by
    \[
    \tilde{\mathcal{J}}(d_i) = \frac{1}{n-m+1} \sum_{j=m}^{n} \varphi \left( |\gamma_i(j) - 2\gamma_{i-1}(j) + \gamma_{i-2}(j)| + \beta |\gamma_i(j) - \gamma_{i-1}(j)| \right)
    \]
    where \( \varphi \) is edge preserving and \( \beta \geq 0 \)
• Step 3 is the same
• Add step 5: classical denoising of \( \hat{f} \) (e.g. shrinkage estimation)
Our method (+curvelets)
Our dejitter: $\varphi(t) = |t|^{0.5}, \beta = 0$

Denoising: curvelets shrinkage
Jitter 10dB SNR+Sin-jitter, $M = 6$

Dejitter: $\varphi(t) = |t|^{0.5}, \beta = 0$

Denoise: curvelets shrinkage
7. Conclusions and perspectives

• Very fast and simple dejittering method yielding remarkable results
• A better exploration for the parameters in presence of noise is necessary
• The case of general impairments + jitter is unexplored
• Go further to restore full jittered sequences (on-going)


see also: [http://www.cmla.ens-cachan.fr/~nikolova/](http://www.cmla.ens-cachan.fr/~nikolova/), Journal papers

Acknowledgements

• the French *Agence Nationale de la Recherche* (ANR), under grant FREEDOM (ANR07-JCJC-0048-01), ”Films, REstauration Et DOnnées Manquantes”.

• *Louis LABORELLI* (Institut Nat. de l’Audiovisuel, France) for discussion on practical questions.

• Dr. *Jackie SHEN* (Manager at Barclays Capital, Wall Street) who gave me his Matlab codes.

• Dr. *Sung-Ha KANG*, Ass. Prof., Georgia Institute of Technology, Atlanta, who realized comparative experiments for me!