AN AUGMENTED LAGRANGIAN METHOD FOR VIDEO RESTORATION

Stanley H. Chan [1] ** , Ramsin Khoshabeh* [1]*, Kristofor B. Gibson* [1]*, Philip E. Gill* [2] *and Truong Q. Nguyen* [1]

University of California, San Diego
[1]Department of Electrical and Computer Engineering
[2]Department of Mathematics

ABSTRACT

This paper presents a fast algorithm for restoring video sequences. The proposed algorithm, as opposed to existing methods, does not consider video restoration as a sequence of image restoration problems. Rather, it treats a video sequence as a space-time volume and poses a space-time total variation regularization to enhance the smoothness of the solution. The optimization problem is solved by transforming the original unconstrained minimization problem to an equivalent constrained minimization problem. An augmented Lagrangian method is used to handle the constraints, and an alternating direction method (ADM) is used to iteratively find solutions of the subproblems. The proposed algorithm has a wide range of applications, including video deblurring and denoising, disparity map refinement, and reducing hot-air turbulence effects.

Index Terms— video, augmented Lagrangian method, alternating direction method, space-time, deconvolution.

1. INTRODUCTION

Solving a video problem is not the same as solving a sequence of image problems. A few unique features about video problems are:

1. Motion information:
   Trajectories of objects/camera in a scene are crucial for motion deblurring. Although estimating trajectories from a single image is possible (e.g., [1]), the methods are limited to camera motion only, or at most one to two objects in a scene. Additionally, these methods rely heavily on object segmentation algorithms - any error in object segmentation will be amplified in the deblurring step. Compared to single images, estimating motion from videos are much easier.
2. Spatial variance versus spatial invariance:
   The motion blur problem is spatially variant if it is considered as a single image problem. A simple example is a scene consisting of a fast moving car and some background buildings. The background is sharp because it is stationary, but the car is motion-blurred because it is moving. There are at least two different blur kernels for this particular problem, and which kernel to use depends on which pixel we are considering. However, if we consider motion blur as a video problem, it becomes spatially *invariant*. It can be shown that a motion blur kernel for a video is a lowpass filter in *time* [2], i.e., independent of the pixel locations. This makes the problem much easier to solve, because invariant blur kernels can be handled efficiently using Fourier Transforms.
3. Temporal consistency:
   Consistency from frame to frame is not guaranteed by performing single image restorations on each frame. As an example, we can create a video sequence by repeating an image many times, with the average intensity of each frame being increased randomly. For this video, there is no noticeable problem if we only look at one frame. However, when it is played, the random fluctuation in average intensity will cause the perception of flickering.

Contact Email: S. Chan, h5chan@ucsd.edu

S. Chan thanks Croucher Foundation Scholarship for supporting this research.

The objective of this paper is to extend a commonly used single image restoration technique to video problems, namely the ADM for total variation minimization problems. We will show that by considering the video as a space-time volume (instead of individual images) [2], the three above mentioned problems can be handled. We will demonstrate practical applications in de-weathering and stereo video processing.

The organization of the paper is as follows. First in Section 2, the problem statement, notation and background are presented. Then in Section 3, the proposed algorithm will be discussed in detail. Section 4 presents three applications of the algorithm. The algorithm's limitation will be mentioned in Section 5.

2. PROBLEM STATEMENT

2.1. Notation and background

A video signal is represented by $f(x, y, t)$, where (x, y) is the spatial coordinate and t is the temporal coordinate. Discrete samples of $f(x, y, t)$ for $x = 0, \ldots, M-1$, $y = 0, \ldots, N-1$, and $t = 0, \ldots, K-1$ form a three-dimensional tensor of M rows, N columns and K frames. This tensor is known as the space-time volume [2].

For the purpose of discussing numerical algorithms, we use matrices and vectors. To this end, we stack the entries of $f(x, y, t)$ into a column vector of dimension $MNK \times 1$, according to lexicographic order. We use the bold letter $\mathbf{f}$ to represent the vectorized version of the space-time volume $f(x, y, t)$. That is, $\mathbf{f} = \mathbf{vec}[f(x, y, t)]$, where $\mathbf{vec}[\cdot]$ denotes the vectorization operator.

Similar image restoration problems, we need a blur kernel to model the blurring operation. The blur kernel in this paper is represented by $h(x, y, t)$. Associated with $h(x, y, t)$ is the convolution matrix $\mathbf{H}$, whose operation on $\mathbf{f}$ is given by $\mathbf{Hf} = \mathbf{vec}[h(x, y, t) * f(x, y, t)]$, where $*$ denotes convolution. If $h(x, y, t)$ is spatially and temporally invariant, then $\mathbf{H}$ is a *triple* block-circulant matrix (analogous to block-circulant-with-circulant-block matrices (BCCB) [3] for image problems). In this case, $\mathbf{H}$ can be diagonalized using the three-dimensional discrete Fourier Transform (3D-DFT) matrix

$\mathbf{F} = F_M \otimes F_N \otimes F_K$, where F_N is the $N \times N$ one-dimensional DFT matrix [4] and $\otimes$ denotes the Kronecker product.

2.2. Main problem

We consider the following least-squares minimization with total variation regularization:

$$\underset{\mathbf{f}}{\text{minimize}} \quad \tfrac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{Df}\|_2\,, \tag{1}$$

where the constant μ is the regularization parameter that puts relative emphasis on the objective function and the regularization term. The vector $\mathbf{f}$ is the optimization variable of size $MNK \times 1$; $\mathbf{g}$ is a vector denoting the observed (blurry and noisy) video. $\mathbf{H}$ is the $MNK \times MNK$ convolution matrix [1]. The norm $\|\cdot\|^2$ is the square of the vector Euclidean norm, and $\|\cdot\|_2$ is defined by (2). The operator $\mathbf{D}$ is the three-dimensional forward finite difference operator, given by $\mathbf{D} = \begin{bmatrix}\mathbf{D}_x^T & \mathbf{D}_y^T & \mathbf{D}_t^T\end{bmatrix}^T$, where

$$\begin{aligned}\mathbf{D}_x\mathbf{f} &= \mathbf{vec}[f(x+1,y,t) - f(x,y,t)],\\ \mathbf{D}_y\mathbf{f} &= \mathbf{vec}[f(x,y+1,t) - f(x,y,t)],\\ \mathbf{D}_t\mathbf{f} &= \mathbf{vec}[f(x,y,t+1) - f(x,y,t)],\end{aligned}$$

with circular boundary conditions. In order to have greater flexibility in controlling the regularization terms, we introduce three scaling factors to the forward difference operators as follows. We define the scalars β_x, β_y and β_t and multiply them with $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$, respectively so that $\mathbf{D} = \begin{bmatrix}\beta_x\mathbf{D}_x^T & \beta_y\mathbf{D}_y^T & \beta_t\mathbf{D}_t^T\end{bmatrix}^T$.

In this paper, $\|\mathbf{Df}\|_2$ is defined as

$$\|\mathbf{Df}\|_2 = \sum_i \sqrt{\beta_x^2\left[\mathbf{D}_x\mathbf{f}\right]_i^2 + \beta_y^2\left[\mathbf{D}_y\mathbf{f}\right]_i^2 + \beta_t^2\left[\mathbf{D}_t\mathbf{f}\right]_i^2}, \tag{2}$$

where $[\mathbf{f}]_i$ is the i-th element of $\mathbf{f}$. When $\beta_x = \beta_y = 1$ and $\beta_t = 0$, then $\|\mathbf{Df}\|_2 = \sum_i \sqrt{[\mathbf{D}_x\mathbf{f}]_i^2 + [\mathbf{D}_y\mathbf{f}]_i^2}$ is the two-dimensional total variation of $\mathbf{f}$ (in space). When $\beta_x = \beta_y = 0$ and $\beta_t = 1$, then $\|\mathbf{Df}\|_2 = \|\mathbf{D}_t\mathbf{f}\|_1$ is the one-dimensional total variation of $\mathbf{f}$ (in time). By adjusting the relative weights β_x, β_y and β_t, we can control the relative emphasis put on the individual terms $\mathbf{D}_x\mathbf{f}$, $\mathbf{D}_y\mathbf{f}$ and $\mathbf{D}_t\mathbf{f}$.

3. PROPOSED ALGORITHM

The proposed algorithm is an extension of the augmented Lagrangian method described in [5], [6]. Therefore, instead of repeating them here, we focus on the changes made to the three-dimensional case. Also, due to limited space, our discussion is focused on Problem (1), which is known as the TV/L2 problem. The same idea can be extended to the TV/L1 problem,

$$\underset{\mathbf{f}}{\text{minimize}}\ \mu\,\|\mathbf{Hf}-\mathbf{g}\|_1 + \|\mathbf{Df}\|_2,$$

by introducing additional intermediate variables $\mathbf{r} = \mathbf{Hf} - \mathbf{g}$ and Lagrange multipliers $\mathbf{z}$ associated with the constraints $\mathbf{r} = \mathbf{Hf} - \mathbf{g}$.

[1] $\mathbf{H}$ is never constructed explicitly, but considered as an operator.

3.1. Overall algorithm

To solve Problem (1), we first introduce intermediate variables $\mathbf{u}$ and transform Problem (1) into an equivalent constrained problem

$$\begin{aligned}\underset{\mathbf{f},\mathbf{u}}{\text{minimize}} \quad & \tfrac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{u}\|_2\\ \text{subject to} \quad & \mathbf{u} = \mathbf{Df}.\end{aligned} \tag{3}$$

The augmented Lagrangian function of Problem (3) is

$$L(\mathbf{f},\mathbf{u},\mathbf{y}) = \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{u}\|_2 - \mathbf{y}^T(\mathbf{u}-\mathbf{Df}) + \frac{\rho_r}{2}\|\mathbf{u}-\mathbf{Df}\|^2, \tag{4}$$

where ρ_r is a regularization parameter associated with the quadratic penalty term $\|\mathbf{u}-\mathbf{Df}\|^2$, and $\mathbf{y}$ are the Lagrange multipliers associated with the constraints $\mathbf{u} = \mathbf{Df}$. In Equation (4), the intermediate variables $\mathbf{u}$ and the Lagrange multipliers $\mathbf{y}$ can be partitioned as

$$\mathbf{u} = \begin{bmatrix}\mathbf{u}_x^T & \mathbf{u}_y^T & \mathbf{u}_t^T\end{bmatrix}^T, \quad \text{and} \quad \mathbf{y} = \begin{bmatrix}\mathbf{y}_x^T & \mathbf{y}_y^T & \mathbf{y}_t^T\end{bmatrix}^T, \tag{5}$$

respectively, and $\|\mathbf{u}\|_2 = \sum_i \sqrt{[\mathbf{u}_x]_i^2 + [\mathbf{u}_y]_i^2 + [\mathbf{u}_t]_i^2}$.

The idea of the augmented Lagrangian method is to find a saddle point of the augmented Lagrangian function $L(\mathbf{f},\mathbf{u},\mathbf{y})$. To this end, we use the alternating direction method (ADM) to solve the following sub-problems iteratively:

$$\begin{aligned}\mathbf{f}_{k+1} &= \underset{\mathbf{f}}{\text{argmin}}\ \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \mathbf{y}_k^T\mathbf{Df} + \frac{\rho_r}{2}\|\mathbf{u}_k - \mathbf{Df}\|^2,\\ \mathbf{u}_{k+1} &= \underset{\mathbf{u}}{\text{argmin}}\ \|\mathbf{u}\|_2 - \mathbf{y}_k^T\mathbf{u} + \frac{\rho_r}{2}\|\mathbf{u} - \mathbf{Df}_{k+1}\|^2,\\ \mathbf{y}_{k+1} &= \mathbf{y}_k - \rho_r(\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}).\end{aligned}$$

We now consider these sub-problems individually.

3.2. f-subproblem

The $\mathbf{f}$-subproblem can be expressed as

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \mathbf{y}^T\mathbf{Df} + \frac{\rho_r}{2}\|\mathbf{u}-\mathbf{Df}\|^2,$$

where its solution can be found by considering the normal equation

$$(\mu\mathbf{H}^T\mathbf{H} + \rho_r\mathbf{D}^T\mathbf{D})\mathbf{f} = \mu\mathbf{H}^T\mathbf{g} + \rho_r\mathbf{D}^T\mathbf{u} - \mathbf{D}^T\mathbf{y}. \tag{6}$$

Since the convolution matrix $\mathbf{H}$ is a triple block-circulant matrix, it is diagonalizable using the 3D-DFT matrix. Hence, the matrix $\mu\mathbf{H}^T\mathbf{H} + \rho_r\mathbf{D}^T\mathbf{D}$ can be diagonalized as

$$\begin{aligned}&\mathbf{F}\left(\mu\mathbf{H}^T\mathbf{H} + \rho_r\mathbf{D}^T\mathbf{D}\right)\mathbf{F}^H\\ &\quad = \mu|\mathbf{\Lambda_H}|^2 + \rho_r(\beta_x^2|\mathbf{\Lambda}_{\mathbf{D}_x}|^2 + \beta_y^2|\mathbf{\Lambda}_{\mathbf{D}_y}|^2 + \beta_t^2|\mathbf{\Lambda}_{\mathbf{D}_t}|^2),\end{aligned} \tag{7}$$

where $\mathbf{\Lambda_H}$, $\mathbf{\Lambda}_{\mathbf{D}_x}$, $\mathbf{\Lambda}_{\mathbf{D}_y}$, $\mathbf{\Lambda}_{\mathbf{D}_t}$ are eigenvalue matrices of $\mathbf{H}$, $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$, respectively. Here, $|\cdot|$ denotes the component-wise complex modulus. With the diagonalization (7), the normal equation (6) can be solved in three steps:

1. Apply three-dimensional discrete Fourier Transform to the right hand side of (6);
2. Multiply the result of Step 1 by $\left(\mu|\mathbf{\Lambda_H}|^2 + \rho_r(\beta_x^2|\mathbf{\Lambda}_{\mathbf{D}_x}|^2 + \beta_y^2|\mathbf{\Lambda}_{\mathbf{D}_y}|^2 + \beta_t^2|\mathbf{\Lambda}_{\mathbf{D}_t}|^2)\right)^{-1}$, which is a component-wise division;
3. Apply an inverse three-dimensional discrete Fourier Transform to the result of Step 2.

Note that the diagonal matrices $\mathbf{\Lambda_H}$, $\mathbf{\Lambda}_{\mathbf{D}_x}$, $\mathbf{\Lambda}_{\mathbf{D}_y}$, $\mathbf{\Lambda}_{\mathbf{D}_t}$ can be precalculated outside the main loop.

3.3. u-subproblem

The $\mathbf{u}$ sub-problem $\min_{\mathbf{u}} \|\mathbf{u}\|_2 - \mathbf{y}^T\mathbf{u} + \frac{\rho_r}{2}\|\mathbf{u} - \mathbf{Df}\|^2$ can be solved using the well-known shrinkage formula [7]. Letting $\mathbf{v}_x = \beta_x\mathbf{D}_x\mathbf{f} + \frac{1}{\rho_r}\mathbf{y}_x$, $\mathbf{v}_y = \beta_y\mathbf{D}_y\mathbf{f} + \frac{1}{\rho_r}\mathbf{y}_y$, $\mathbf{v}_t = \beta_t\mathbf{D}_t\mathbf{f} + \frac{1}{\rho_r}\mathbf{y}_t$, and let $\mathbf{v} = \sqrt{|\mathbf{v}_x|^2 + |\mathbf{v}_y|^2 + |\mathbf{v}_t|^2}$, $\mathbf{u}$ can be found as

$$\begin{aligned}\mathbf{u}_x &= \max\{\mathbf{v} - 1/\rho_r, 0\} \cdot \mathbf{v}_x/\mathbf{v},\\ \mathbf{u}_y &= \max\{\mathbf{v} - 1/\rho_r, 0\} \cdot \mathbf{v}_y/\mathbf{v},\\ \mathbf{u}_t &= \max\{\mathbf{v} - 1/\rho_r, 0\} \cdot \mathbf{v}_t/\mathbf{v},\end{aligned} \tag{8}$$

where the multiplications and divisions are component-wise operations.

3.4. Parameters

The regularization parameter ρ_r associated with the quadratic penalty $\|\mathbf{u} - \mathbf{Df}\|^2$ is updated according to the relative change between the previous constraint violation and the current constraint violation. Specifically, given an initial ρ_r (typically $\rho_r = 2$), ρ_r is updated by

$$\rho_r = \begin{cases} 2\rho_r, & \text{if } \|\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}\|^2 \geq \alpha_r\|\mathbf{u}_k - \mathbf{Df}_k\|^2,\\ \rho_r, & \text{otherwise.}\end{cases} \tag{9}$$

Empirically, $\alpha_r = 0.7$ is an appropriate choice for most of the problems.

A few remarks for choosing the parameter μ and $(\beta_x, \beta_y, \beta_t)$:

- Regularization parameter μ
  The range of μ is from 1 to 10^6, where smaller values of μ will produce smoother solutions.
- Relative weights $(\beta_x, \beta_y, \beta_t)$
  For gray-scaled/ color image restoration, $(\beta_x, \beta_y, \beta_t) = (1, 1, 0)$. For general video deblurring and denoising, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$. For removing serious flickering in video, $(\beta_x, \beta_y, \beta_t) = (1, 1, 10)$.

3.5. Convergence and Run Time

The convergence proof for conventional ADM can be found in [8, 6], in which the parameter ρ_r is fixed throughout the iterations. When the automatic parameter update scheme (9) is used, the algorithm converges at a faster rate of convergence. The complexity of the algorithm has the bottleneck in solving the normal equation (6), which requires $O(n \log n)$ operations for the three-dimensional discrete Fourier Transforms, with n being the number of elements of the space-time volume $f(x, y, t)$.

The run time of the proposed algorithm is approximately 2 seconds per frame per color channel on Intel Qual Core 2.8GHz/ 4GB RAM/ MATLAB/ Windows 7, for a 300×400 video. Therefore, some low frame-rate applications (5-10 frames per second) can be supported when the algorithm is ported to CPU or GPU.

3.6. User Interface

The proposed algorithm (`deconvtv`) has been implemented in MATLAB/ Windows 7. The name and the interface are designed to match MATLAB's standard deconvolution tools, namely `deconvwnr`, `deconvwnr` and `deconvreg`. The i/o format of `deconvtv` in MATLAB is

```
out = deconvtv(g, H, mu, opts);
```

where `g`, `H` and `mu` are the required fields, and `opts` is an optional field that controls the parameters.

4. APPLICATIONS

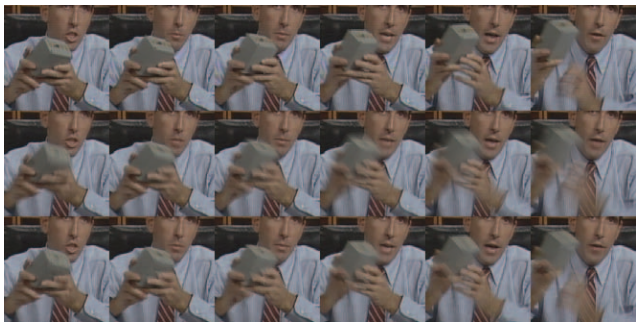
We illustrate a few applications of the algorithm in this section. Due to the limited space, more results can be found at `http://videoprocessing.ucsd.edu/~stanleychan/deconvtv`.

4.1. Video Deblurring

Our first example is video deblurring. As mentioned in [2], motion blur can be considered as the convolution of the space-time volume $f(x, y, t)$ with the three-dimensional blur kernel $h(x, y, t)$, where $h(x, y, t)$ is defined as $h(x, y, t) = 1/T$ for $0 < t < T$, $x = y = 0$, and $h(x, y, t) = 0$ otherwise. Consequently, given an observed video $g(x, y, t)$ and the blur kernel $h(x, y, t)$, we solve the following minimization problem

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{Hf} - \mathbf{g}\|^2 + \|\mathbf{Df}\|_2.$$

To illustrate the performance of the proposed method, we tested a number of video sequences. The video sequences are blurred by a point spread function $h(t) = 1/4$ for $t = 0, 1, 2, 3$, and $h(t) = 0$ otherwise. All frames are corrupted with Gaussian noise so that the blur signal to noise ratio (BSNR) is 30dB. Some results are shown in Fig. 1. For more sequences, PSNR values, total variation values in space and time, please visit our website.



frame no. $\rightarrow$

Fig. 1. Motion blur experiment for "salesman" sequence. Top: original video sequence. Middle: motion blurred by $h(t) = 1/4$ for $t = 0, 1, 2, 3$. Bottom: video recovered by `deconvtv`. Parameters for `deconvtv` are $\mu = 1000$, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$.

4.2. Disparity Refinement

Our second example deals with the refinement of the disparity maps estimated from a stereo video. Disparity is the reciprocal of the distance between the camera and the object (i.e., the depth). Disparity estimation is the first step to all stereo video processing applications, such as object detection in three-dimensional space, saliency for stereo videos, stereo coding etc. However, most of the existing methods [9] are only applicable to images, not videos. The temporal consistency of the results generated by those methods is usually very poor.

Given a stereo video, we first perform initial disparity map estimation in a frame-by-frame basis using a modified version of loopy

belief propagation [10]. Then we consider the initial disparity maps as a space-time volume $\mathbf{g}$ and solve the following minimization problem

$$\underset{\mathbf{f}}{\text{minimize}} \quad \mu \|\mathbf{f}-\mathbf{g}\|_1 + \|\mathbf{D}\mathbf{f}\|_2.$$

The goal of this denoising problem is to reduce the temporal and spatial noise in the disparity, while keeping the edges sharp. The choice of l_1-norm over the l_2-norm in the objective function is based on two reasons. First, the noise in the initial disparities is impulsive, which is more suitable for using TV/L1, as TV/L1 minimization enforces sparsity of $\mathbf{f}-\mathbf{g}$, which in turn suppresses impulsive noise. Second, disparity maps consist of only a few discrete levels. So the solution $\mathbf{f}$ is a piece-wise step function, which also makes TV/L1 minimization more suitable than TV/L2 minimization.

Fig. 2 shows zoom-in results of the sequence "Old Timers". More sequences and plots can be found on our website.

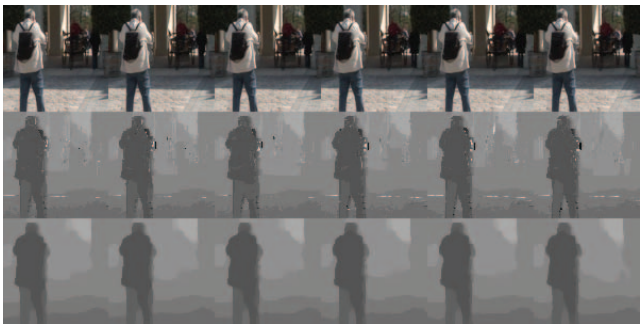


Fig. 2. Disparity map estimation result of "Old Timers" sequence. Top row: stereo video input (left view). Middle row: initial map estimation using loopy belief propagation [10]. Bottom row: result of applying `deconvtv` to the middle row. Parameters are $\mu = 1$, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$.

4.3. Hot-air Turbulence

Our third example is to remove hot-air turbulence effects. In the presence of hot-air turbulence, the refractive index along the transmission path of the light ray is changing [11] and so the path differences and hence the phases are varying. Consequently, the observed image is distorted by geometric warping, motion blur and sometimes out-of-focus blur.

The objective of using `deconvtv` is to reduce the turbulence effects, and deblur the video. To do so, we consider the video as a space-time volume. For a particular pixel in space, pixel intensity fluctuates rapidly in time if it is distortion by hot-air turbulence. Therefore, by setting β_t large, the variation of pixel intensity in time will be suppressed, and hence the turbulence effects will be reduced. Hence, we solve the minimization problem

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{H}\mathbf{f}-\mathbf{g}\|^2 + \|\mathbf{D}\mathbf{f}\|_2,$$

where $\mathbf{H}$ is estimated using a modified version of the kernel estimation method presented in [12]. Some results are shown in Fig. 3, and full resolution videos/ other sequences can be found on our website.

5. DISCUSSION AND CONCLUSION

A limitation of `deconvtv` in solving motion blur problems is that a robust frame rate up conversion algorithm is needed, because the performance of the algorithm is sensitive to the quality of intermediate frames. Another limitation of `deconvtv` is that it handles small hot-air turbulence only (perceptually similar to jittering). For large area geometric distortion, non-rigid registration is needed.

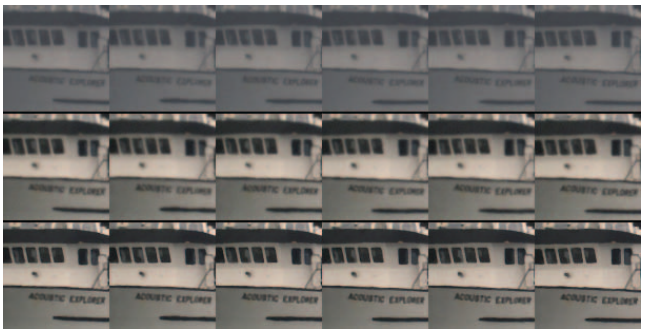


Fig. 3. Zoom-in snapshots of the sequence "acoustic explorer". Top: input video sequence. Middle: contrast enhancement using gray-level grouping [13]. Bottom: applying `deconvtv` to the results of the middle row.

To conclude, we propose and demonstrate a fast numerical optimization algorithm that solves video restoration problems. The algorithm treats the video as a space-time volume, instead of a sequence of individual images. Therefore, spatial and temporal smoothness can be enforced by setting up a total variation regularization. Our algorithm can be used in several areas, including video deblurring/ denoising, disparity refinement and hot-air turbulence removal.

6. REFERENCE

[1] S. Dai and Y. Wu, "Motion from blur," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, 2008.

[2] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 27, pp. 531–545, 2005.

[3] B. Kim, *Numerical Optimization Methods for Image Restoration*, Ph.D. thesis, Stanford University, Dec 2002.

[4] M. K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, Inc, 2004.

[5] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345 – 2356, Sept 2010.

[6] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," Tech. Rep., CAAM, Rice University, Sep 2008.

[7] C. Li, *An Efficient Algorithm For Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing*, Master thesis, Rice University, 2009, available at `http://www.caam.rice.edu/~optimization/L1/TVAL3/tval3_thesis.pdf`.

[8] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[9] "Middleburry stereo dataset," available at `http://bj.middlebury.edu/~schar/stereo/web/results.php`.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, 2006.

[11] M. C. Roggemann and B. M. Welsh, *Imaging Through Turbulence*, CRC Press, 1996.

[12] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *European Conference on Computer Vision (ECCV' 10)*, 2010, available at `http://www.cse.cuhk.edu.hk/~leojia/projects/robust_deblur/`.

[13] Z. Chen, B. R. Abidi, D. L. Page, and M. A. Abidi, "Gray-level grouping (GLG): an automatic method for optimized image contrast enhancement-part II," *IEEE Transactions on Image Processing*, vol. 15, pp. 2303 – 2314, 2006.