An Augmented Lagrangian Method for Total Variation Video Restoration

Stanley H. Chan, Ramsin Khoshabeh, Kristofor B. Gibson, *Student Members, IEEE,* Philip E. Gill, and Truong Q. Nguyen, *Fellow, IEEE.*

***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, video disparity refinement, and hot-air turbulence effect reduction.**

***Index Terms*—Augmented Lagrangian, total variation, alternating direction method, video restoration, video deblurring, video disparity, hot-air turbulence**

I. Introduction

A. Video Restoration Problems

Image restoration is an inverse problem where the objective is to recover a sharp image from a blurry and noisy observation. Mathematically, a linear shift invariant imaging system is modeled as [1]

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \eta, \tag{1}$$

where $\mathbf{f} \in \mathbb{R}^{MN\times 1}$ is a vector denoting the unknown (potentially sharp) image of size $M \times N$, $\mathbf{g} \in \mathbb{R}^{MN\times 1}$ is a vector denoting the observed image, $\eta \in \mathbb{R}^{MN\times 1}$ is a vector denoting the noise, and the matrix $\mathbf{H} \in \mathbb{R}^{MN\times MN}$ is a linear transformation representing convolution operation. The goal of image restoration is to recover $\mathbf{f}$ from the observed image $\mathbf{g}$.

Standard single image restoration has been studied for more than half century. Popular methods such as Wiener deconvolution [1], Lucy Richardson deconvolution [2], [3] and regularized least squares minimization [4], [5] have already been implemented in MATLAB and FIJI [6]. Advanced methods such as total variational image restoration methods are also becoming mature [7]–[11].

While single image restorations still have room for improvement, we consider in this paper the video restoration problem. The key difference between image and video is the additional time dimension. Consequently, video restoration has some unique features that do not exist in image restoration:

1) Motion information
   Motion deblurring requires motion vector field, which can be estimated from a video sequence using conventional methods such as block-matching [12] and optical flow [13]. While it is also possible to remove motion blur based on a single image, for example, [14]–[18], the performance is limited to global motion or at most one to two objects by using sophisticated object segmentation algorithms.
2) Spatial variance versus spatial invariance
   For a class of spatially variant image restoration problems (in particular motion blur), the convolution matrix $\mathbf{H}$ is not a block-circulant matrix. Therefore, Fourier Transforms cannot be utilized to efficiently find a solution. Videos, in contrast, allow us to transform a sequence of spatially variant problems to a spatially *invariant* problem (See the next section for more discussions). As a result, huge gain in speed can be realized.
3) Temporal consistency
   Temporal consistency concerns about the smoothness of the restored video along the time axis. Although smoothing can be performed spatially (as in the case of single image restoration), temporal consistency cannot be guaranteed if these methods are applied to a video in a frame-by-frame basis.

Because of these unique features in video, we seek a video restoration algorithm that utilizes motion information, exploits the spatially invariant properties and enforces spatial and temporal consistency.

B. Related Work

Majority of existing video restoration methods recover a video by solving a sequence of individual image restorations. To improve the temporal consistency among the frames, various approaches can be adopted: [19] modified Equation (1) to incorporate the geometric warp caused by motion; [20] utilized the motion vector field as a prior to the restoration; [21] considered a regularization function of the residue between the current solution and the motion compensated version of the previous solution.

Another class of methods are based on the concept of “space-time volume”, which is first introduced in the early 90’s by Jähne [22], and rediscovered by Wexler, Shechtman,

S. Chan, R. Khoshabeh, K. Gibson and T. Nguyen are with the Department of Electrical and Computer Engineering, University of California, San Diego.

P. Gill is with the Department of Mathematics, University of California, San Diego.

Contact author: S. Chan, h5chan@ucsd.edu. The work of S. Chan is supported by Croucher Foundation Scholarship, Hong Kong.

Manuscript received XXX

Caspi and Irani [23], [24]. The idea of space-time volume is to stack the frames of a video to form a three-dimensional data structure known as the space-time volume. This allows one to transform the spatially variant motion blur problem to a spatially invariant problem. By imposing regularization functions along the spatial and temporal directions respectively, both spatial and temporal smoothness can be enforced. However, size of a space-time volume is much larger than a single image. Therefore, the authors of [24] only consider a Tikhonov regularized least-squares minimization (Equation (3) of [24]) as there is a closed formed solution. More sophisticated regularization functions such as total variation [19] and its l_1-approximation [25] do not seem possible under this framework, for these non-differentiable functions are difficult to solve efficiently.

This paper investigates the total variation regularization functions in space-time minimization. In particular, we consider the following two problems:

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{f}\|_{TV}, \tag{2}$$

which is known as the TV/L2 minimization, and

$$\underset{\mathbf{f}}{\text{minimize}} \quad \mu\|\mathbf{Hf}-\mathbf{g}\|_1 + \|\mathbf{f}\|_{TV}, \tag{3}$$

which is known as the TV/L1 minimization. Unless specified, the norms $\|\cdot\|^2$ and $\|\cdot\|_1$ are the conventional vector 2-norm squares and the vector 1-norm, respectively. The total variation (TV)-norm $\|\mathbf{f}\|_{TV}$ can either be the anisotropic total variation norm

$$\|\mathbf{f}\|_{TV1} = \sum_i \left(\beta_x|[\mathbf{D}_x\mathbf{f}]_i| + \beta_y|[\mathbf{D}_y\mathbf{f}]_i| + \beta_t|[\mathbf{D}_t\mathbf{f}]_i|\right), \tag{4}$$

or the isotropic total variation norm

$$\|\mathbf{f}\|_{TV2} = \sum_i \sqrt{\beta_x^2[\mathbf{D}_x\mathbf{f}]_i^2 + \beta_y^2[\mathbf{D}_y\mathbf{f}]_i^2 + \beta_t^2[\mathbf{D}_t\mathbf{f}]_i^2}, \tag{5}$$

where the operators $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ are the forward finite difference operators along the horizontal, vertical and temporal directions, respectively. Here, $(\beta_x, \beta_y, \beta_t)$ are constants, and $[\mathbf{f}]_i$ denotes the i-th component of the vector $\mathbf{f}$. More details on these two equations will be discussed in Section II.D.

The proposed algorithm is based on the augmented Lagrangian method, an old method that recently draws significant attentions [10], [11], [26]. All of these methods follow from Eckstein and Bertsekas's operator splitting method [27], which can be traced back to the work by Douglas and Rachford [28], and the proximal point algorithm by Rockafellar [29], [30]. Recently, the operator splitting method has been proven to be equivalent to the splitting Bregman iteration for some problems [31], [32]. However, there is no work on extending the augmented Lagrangian method to space-time minimization.

C. Contributions

The contribution of this paper is summarized as follows:

- We extend the existing augmented Lagrangian method to solve space-time total variation minimization problems (2) and (3).
- In terms of restoration quality, our method achieves TV/L1 and TV/L2 minimization quality. The quality is significantly better than its counter part [24] which is a space-time Tikhonov least-squares minimization. The proposed method also gives better results than a number of video restoration algorithms that use motion compensation.
- In terms of speed, we achieve significantly higher computational speed compared to existing methods. Typical run time to deblur and denoise a 300×400 gray-scaled video is a few second per frame on a personal computer (MATLAB). This implies the possibility of real-time processing on GPU.
- In terms of class of problems, we are able to remove spatially variant motion blur, whereas existing frame-by-frame based approaches are either unable to do so, or computationally slow.
- Applications: (1). Video deblurring - With the assistance of frame rate up conversion algorithms, the proposed method can remove spatially variant motion blur for *real* video sequences. (2). Video disparity - Occlusion errors and temporal inconsistent estimates in the video disparity can be handled by the proposed algorithm without any modification. (3). Hot-air turbulence - The algorithm can be directly used to deblur and remove hot-air turbulence effects.

D. Organization

This paper is an extension of two recently accepted conference papers [33], [34]. The organization of this paper is as follows. Section II consists of notations and background materials. The algorithms are discussed in Section III. Section IV discusses three applications of the proposed algorithm, namely (1). Video deblurring (2). Video disparity refinement, (3). Hot-air turbulence effects reduction. A concluding remark is given in Section V.

II. Background and Notation

A. Notation

A video signal is represented by a three-dimensional function $f(x, y, t)$, where (x, y) denotes the coordinate in space and t denotes the coordinate in time. Suppose that each frame of the video has M rows, N columns, and there are K frames, then the 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 size $M \times N \times K$.

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 size $MNK \times 1$, according to the lexicographic order. We use the bold letter $\mathbf{f}$ to represent the vectorized version of the space-time volume $f(x, y, t)$, i.e.,

$$\mathbf{f} = \mathbf{vec}(f(x, y, t)),$$

where $\mathbf{vec}(\cdot)$ represents the vectorization operator.

B. Three-dimensional Convolution

The three-dimensional convolution is a natural extension of the conventional two-dimensional convolution. Given a space-time volume $f(x,y,t)$ and the point spread function $h(x,y,t)$, the convolved signal $g(x,y,t)$ is given by $g(x,y,t) = f(x,y,t)*h(x,y,t) \overset{def}{=} \sum_{u,v,\tau} h(u,v,\tau)f(x-u,y-v,t-\tau)$. Convolution is a linear operation, so it can be expressed using matrices. More precisely, we define the convolution matrix associated with a blur kernel $h(x,y,t)$ as the linear operator that maps the signal $f(x,y,t)$ to $g(x,y,t)$ following the rule

$$\mathbf{Hf} = \mathbf{vec}(g(x,y,t)) = \mathbf{vec}(h(x,y,t) * f(x,y,t)). \tag{6}$$

Assuming periodic boundaries [35], the convolution matrix $\mathbf{H}$ is a *triple* block-circulant matrix - it has a block circulant structure, and within each block there is a block-circulant-with-circulant block (BCCB) submatrix. Circulant matrices are diagonalizable using discrete Fourier Transform matrices [36], [37]:

Fact 1: If $\mathbf{H}$ is a triple block-circulant matrix, then it can be diagonalized by the three-dimensional DFT matrix $\mathbf{F}$:

$$\mathbf{H} = \mathbf{F}^H \mathbf{\Lambda} \mathbf{F},$$

where $(\cdot)^H$ is the Hermitian operator, and $\mathbf{\Lambda}$ is a diagonal matrix storing the eigenvalues of $\mathbf{H}$.

C. Forward Difference Operators

We define an operator $\mathbf{D}$ be a collection of three sub-operators $\mathbf{D} = \begin{bmatrix} \mathbf{D}_x^T & \mathbf{D}_y^T & \mathbf{D}_t^T \end{bmatrix}^T$, where $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ are the first-order forward finite difference operators along the horizontal, vertical and temporal directions, respectively. The definitions of each individual sub-operators are

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

with periodic boundary conditions.

In order to have greater flexibility in controlling the forward difference along each direction, we introduce three scaling factors 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 $\mathbf{D} = \begin{bmatrix} \beta_x\mathbf{D}_x^T & \beta_y\mathbf{D}_y^T & \beta_t\mathbf{D}_t^T \end{bmatrix}^T$.

With $(\beta_x, \beta_y, \beta_t)$, the anisotropic total variation norm $\|\mathbf{f}\|_{TV1}$ and the isotropic total variation $\|\mathbf{f}\|_{TV2}$ are defined according to Equations (4) and (5), respectively. When $\beta_x = \beta_y = 1$ and $\beta_t = 0$, $\|\mathbf{f}\|_{TV2}$ is the two-dimensional total variation of $\mathbf{f}$ (in space). When $\beta_x = \beta_y = 0$ and $\beta_t = 1$, $\|\mathbf{f}\|_{TV2}$ is the one-dimensional total variation of $\mathbf{f}$ (in time). By adjusting β_x, β_y and β_t, we can control the relative emphasis put on individual terms $\mathbf{D}_x\mathbf{f}$, $\mathbf{D}_y\mathbf{f}$ and $\mathbf{D}_t\mathbf{f}$.

Note that $\|\mathbf{f}\|_{TV1}$ is equivalent to the vector 1-norm on $\mathbf{Df}$, i.e., $\|\mathbf{f}\|_{TV1} = \|\mathbf{Df}\|_1$. Therefore, for notation simplicity we use $\|\mathbf{Df}\|_1$ instead. For $\|\mathbf{f}\|_{TV2}$, although $\|\mathbf{f}\|_{TV2} \neq \|\mathbf{Df}\|_2$ using the vector 2-norm definition, we still define $\|\mathbf{Df}\|_2 \overset{def}{=} \|\mathbf{f}\|_{TV2}$ to align with the definition of $\|\mathbf{Df}\|_1$. However, this will be made clear if confusion arises.

III. Proposed Algorithm

The proposed algorithm belongs to the family of operator splitting methods [10], [11], [27]. Therefore, instead of repeating the details, we focus on the modifications made to the three-dimensional data structure. Additionally, our discussion is focused on the anisotropic TV, i.e., $\|\mathbf{Df}\|_1$. The isotropic TV, $\|\mathbf{Df}\|_2$ can be derived similarly.

A. TV/L2 Problem

The core optimization problem that we solve is the following TV/L2 minimization:

$$\underset{\mathbf{f}}{\text{minimize}} \quad \tfrac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{Df}\|_1, \tag{7}$$

where μ is a regularization parameter. To solve Problem (7), we first introduce intermediate variables $\mathbf{u}$ and transform Problem (7) into an equivalent problem

$$\begin{aligned} &\underset{\mathbf{f},\mathbf{u}}{\text{minimize}} && \tfrac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{u}\|_1 \\ &\text{subject to} && \mathbf{u} = \mathbf{Df}. \end{aligned} \tag{8}$$

The augmented Lagrangian of Problem (8) is

$$L(\mathbf{f},\mathbf{u},\mathbf{y}) = \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 + \|\mathbf{u}\|_1 - \mathbf{y}^T(\mathbf{u}-\mathbf{Df}) + \frac{\rho_r}{2}\|\mathbf{u}-\mathbf{Df}\|^2, \tag{9}$$

where ρ_r is a regularization parameter associated with the quadratic penalty term $\|\mathbf{u}-\mathbf{Df}\|^2$, and $\mathbf{y}$ is the Lagrange multiplier associated with the constraint $\mathbf{u} = \mathbf{Df}$. In Equation (9), the intermediate variable $\mathbf{u}$ and the Lagrange multiplier $\mathbf{y}$ can be partitioned as

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_x^T & \mathbf{u}_y^T & \mathbf{u}_t^T \end{bmatrix}^T, \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_x^T & \mathbf{y}_y^T & \mathbf{y}_t^T \end{bmatrix}^T, \tag{10}$$

respectively.

The idea of the augmented Lagrangian method is to find a saddle point of $L(\mathbf{f},\mathbf{u},\mathbf{y})$, which is also the solution of the original problem (7). To this end, we use the alternating direction method (ADM) to solve the following sub-problems iteratively:

$$\mathbf{f}_{k+1} = \underset{\mathbf{f}}{\text{argmin}} \quad \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2 - \mathbf{y}_k^T(\mathbf{u}_k - \mathbf{Df}) + \frac{\rho_r}{2}\|\mathbf{u}_k - \mathbf{Df}\|^2, \tag{11}$$

$$\mathbf{u}_{k+1} = \underset{\mathbf{u}}{\text{argmin}} \quad \|\mathbf{u}\|_1 - \mathbf{y}_k^T(\mathbf{u} - \mathbf{Df}_{k+1}) + \frac{\rho_r}{2}\|\mathbf{u} - \mathbf{Df}_{k+1}\|^2, \tag{12}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \rho_r(\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}). \tag{13}$$

We now investigate these sub-problems one by one.

1) $\mathbf{f}$-subproblem: By dropping the indices k, solution of Problem (11) is found by considering the normal equation

$$(\mu\mathbf{H}^T\mathbf{H} + \rho_r\mathbf{D}^T\mathbf{D})\mathbf{f} = \mu\mathbf{H}^T\mathbf{g} + \rho_r\mathbf{D}^T\mathbf{u} - \mathbf{D}^T\mathbf{y}. \tag{14}$$

The convolution matrix $\mathbf{H}$ in Equation (14) is a triple block-circulant matrix, and therefore by Fact 1, $\mathbf{H}$ can be diagonalized using the 3D-DFT matrix. Hence, (14) has a solution:

$$\mathbf{f} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[\mu\mathbf{H}^T\mathbf{g} + \rho_r\mathbf{D}^T\mathbf{u} - \mathbf{D}^T\mathbf{y}]}{\mu|\mathcal{F}[\mathbf{H}]|^2 + \rho_r(|\mathcal{F}[\mathbf{D}_x]|^2 + |\mathcal{F}[\mathbf{D}_y]|^2 + |\mathcal{F}[\mathbf{D}_t]|^2)}\right], \tag{15}$$

Algorithm 1 Algorithm for TV/L2 minimization problem

```
Input data $\mathbf{g}$ and $\mathbf{H}$.
Input parameters $\mu$, $\beta_x$, $\beta_y$, $\beta_t$.
Set parameters $\rho_r$ (default = 2), $\alpha_0$ (default = 0.7).
Initialize $\mathbf{f}_0 = \mathbf{g}$, $\mathbf{u}_0 = \mathbf{D}\mathbf{f}_0$, $\mathbf{y} = \mathbf{0}$, $k = 0$.
Compute the matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, $\mathcal{F}[\mathbf{H}]$.
while not converge do
  1. Solve the $\mathbf{f}$-subproblem (11) using Equation (15).
  2. Solve the $\mathbf{u}$-subproblem (12) using Equation (16).
  3. Update the Lagrange multiplier $\mathbf{y}$ using Equation (13).
  4. Update $\rho_r$ according to Equation (24).
  5. Check convergence:
  if $\|\mathbf{f}_{k+1} - \mathbf{f}_k\|_2 / \|\mathbf{f}_k\|_2 \le \text{tol}$ then
    break
  end if
end while
```

where $\mathcal{F}$ denotes the three-dimensional Fourier Transform operator. The matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, $\mathcal{F}[\mathbf{H}]$ can be pre-calculated outside the main loop. Therefore, the complexity of solving Equation (14) is in the order of $O(n \log n)$ operations, which is the complexity of the three-dimensional Fourier Transforms and n is the number of elements of the space-time volume $f(x, y, t)$.

2) $\mathbf{u}$-subproblem: Problem (12) is known as the $\mathbf{u}$-subproblem, which can be solved using the shrinkage formula [38]. Letting $\mathbf{v}_x = \beta_x \mathbf{D}_x \mathbf{f} + \frac{1}{\rho_r}\mathbf{y}_x$, (analogous definitions for $\mathbf{v}_y$ and $\mathbf{v}_t$), $\mathbf{u}_x$ is given by

$$\mathbf{u}_x = \max\left\{|\mathbf{v}_x| - \frac{1}{\rho_r}, 0\right\} \cdot \operatorname{sign}(\mathbf{v}_x). \tag{16}$$

Analogous solutions for $\mathbf{u}_y$ and $\mathbf{u}_t$ can also be derived.

In case of isotropic TV, the solution is given by [38]

$$\mathbf{u}_x = \max\left\{\mathbf{v} - \frac{1}{\rho_r}, 0\right\} \cdot \frac{\mathbf{v}_x}{\mathbf{v}}, \tag{17}$$

where $\mathbf{v} = \sqrt{|\mathbf{v}_x|^2 + |\mathbf{v}_y|^2 + |\mathbf{v}_t|^2}$, and the multiplication and divisions are component-wise operations.

3) Algorithm: Algorithm 1 shows the pseudo-code of the TV/L2 algorithm.

B. TV/L1 Problem

TV/L1 problem can be solved by introducing two intermediate variables $\mathbf{r}$ and $\mathbf{u}$, and modify Problem (3) as

$$\begin{aligned} \underset{\mathbf{f},\mathbf{r},\mathbf{u}}{\text{minimize}} \quad & \mu\|\mathbf{r}\|_1 + \|\mathbf{u}\|_1 \\ \text{subject to} \quad & \mathbf{r} = \mathbf{H}\mathbf{f} - \mathbf{g} \\ & \mathbf{u} = \mathbf{D}\mathbf{f}. \end{aligned} \tag{18}$$

The augmented Lagrangian of (18) is given by $L(\mathbf{f}, \mathbf{r}, \mathbf{u}, \mathbf{y}, \mathbf{z}) = \mu\|\mathbf{r}\|_1 + \|\mathbf{u}\|_1 - \mathbf{z}^T(\mathbf{r} - \mathbf{H}\mathbf{f} + \mathbf{g}) + \frac{\rho_o}{2}\|\mathbf{r} - \mathbf{H}\mathbf{f} + \mathbf{g}\|^2 - \mathbf{y}^T(\mathbf{u} - \mathbf{D}\mathbf{f}) + \frac{\rho_r}{2}\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$. Here, the variable $\mathbf{y}$ is the Lagrange multiplier associated with constraint $\mathbf{u} = \mathbf{D}\mathbf{f}$ and the variable $\mathbf{z}$ is the Lagrange multiplier associated with the constraint $\mathbf{r} = \mathbf{H}\mathbf{f} - \mathbf{g}$. Moreover, $\mathbf{u}$ and $\mathbf{y}$ can be partitioned as in Equation (10). The parameters ρ_o and ρ_r are two regularization parameters. The subscripts "o" and "r" stand for "**o**bjective", and "**r**egularization", respectively.

1) $\mathbf{f}$-subproblem: The $\mathbf{f}$-subproblem of TV/L1 is

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\rho_o}{2}\|\mathbf{r} - \mathbf{H}\mathbf{f} + \mathbf{g}\|^2 + \frac{\rho_r}{2}\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2 + \mathbf{z}^T\mathbf{H}\mathbf{f} + \mathbf{y}^T\mathbf{D}\mathbf{f}, \tag{19}$$

which can be solved by considering the normal equation

$$(\rho_o\mathbf{H}^T\mathbf{H} + \rho_r\mathbf{D}^T\mathbf{D})\mathbf{f} = \rho_o\mathbf{H}^T\mathbf{g} + \mathbf{H}^T(\rho_o\mathbf{r} - \mathbf{z}) + \mathbf{D}^T(\rho_r\mathbf{u} - \mathbf{y}),$$

yielding

$$\mathbf{f} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}[\rho_o\mathbf{H}^T\mathbf{g} + \mathbf{H}^T(\rho_o\mathbf{r} - \mathbf{z}) + \mathbf{D}^T(\rho_r\mathbf{u} - \mathbf{y})]}{\rho_o|\mathcal{F}[\mathbf{H}]|^2 + \rho_r(|\mathcal{F}[\mathbf{D}_x]|^2 + |\mathcal{F}[\mathbf{D}_y]|^2 + |\mathcal{F}[\mathbf{D}_t]|^2)}\right]. \tag{20}$$

Algorithm 2 Algorithm for TV/L1 minimization problem

```
Input $\mathbf{g}$, $\mathbf{H}$ and parameters $\mu$, $\beta_x$, $\beta_y$, $\beta_t$. Let $k = 0$.
Set parameters $\rho_r$ (default = 2), $\rho_o$ (default = 100), $\alpha_0$ (default = 0.7).
Initialize $\mathbf{f}_0 = \mathbf{g}$, $\mathbf{u}_0 = \mathbf{D}\mathbf{f}_0$, $\mathbf{y}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{H}\mathbf{f}_0 - \mathbf{g}$, $\mathbf{z}_0 = \mathbf{0}$.
Compute the matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, $\mathcal{F}[\mathbf{H}]$.
while not converge do
  1. Solve the $\mathbf{f}$-subproblem (19) using Equation (20).
  2. Solve the $\mathbf{u}$-subproblem (12) using Equation (16).
  3. Solve the $\mathbf{r}$-subproblem (21) using Equation (22).
  4. Update $\mathbf{y}$ and $\mathbf{z}$ using Equation (23).
  5. Update $\rho_r$ and $\rho_o$ according to Equation (24).
  6. Check convergence:
  if $\|\mathbf{f}_{k+1} - \mathbf{f}_k\|_2 / \|\mathbf{f}_k\|_2 \le \text{tol}$ then
    break
  end if
end while
```

2) $\mathbf{u}$-subproblem: The $\mathbf{u}$-subproblem of TV/L1 is the same as that of TV/L2. Therefore, the solution is given by Equation (16).

3) $\mathbf{r}$-subproblem: The $\mathbf{r}$-subproblem is

$$\underset{\mathbf{r}}{\text{minimize}} \quad \mu\|\mathbf{r}\|_1 - \mathbf{z}^T\mathbf{r} + \frac{\rho_o}{2}\|\mathbf{r} - \mathbf{H}\mathbf{f} + \mathbf{g}\|^2. \tag{21}$$

Thus using the shrinkage formula, the solution is

$$\mathbf{r} = \max\left\{\left|\mathbf{H}\mathbf{f} - \mathbf{g} + \frac{1}{\rho_o}\mathbf{z}\right| - \frac{\mu}{\rho_o}, 0\right\} \cdot \operatorname{sign}\left(\mathbf{H}\mathbf{f} - \mathbf{g} + \frac{1}{\rho_o}\mathbf{z}\right). \tag{22}$$

4) Multiplier update: $\mathbf{y}$ and $\mathbf{z}$ are updated as

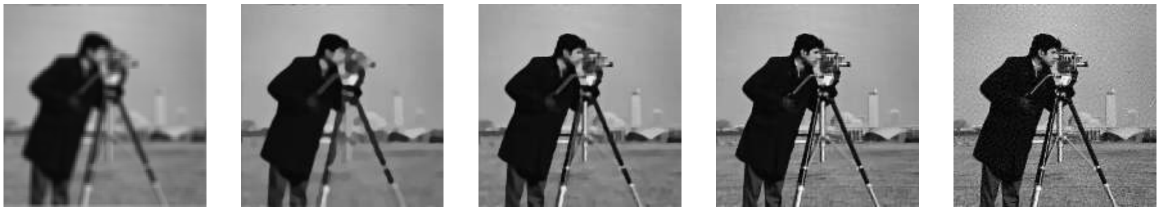
$$\begin{aligned} \mathbf{y}_{k+1} &= \mathbf{y}_k - \rho_r(\mathbf{u}_{k+1} - \mathbf{D}\mathbf{f}_{k+1}), \\ \mathbf{z}_{k+1} &= \mathbf{z}_k - \rho_o(\mathbf{r}_{k+1} - \mathbf{H}\mathbf{f}_{k+1} + \mathbf{g}). \end{aligned} \tag{23}$$

5) Algorithm: Algorithm 2 shows the pseudo-code of the TV/L1 algorithm.

C. Parameters

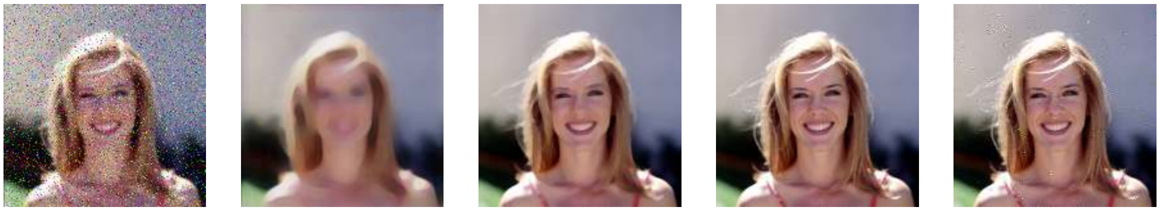
In this subsection we discuss the choice of parameters.

1) Choosing μ: The regularization parameter μ trades off the least-squares error and the total variation penalty. Large values of μ tend to give sharper results, but noise will be amplified. Small values of μ give less noisy results, but the image may be smoothed. The choice of μ is not known



Input $\mu = 10^2$ $\mu = 10^3$ $\mu^* = 10352$ $\mu = 10^5$

Fig. 1. TV/L2 Image recovery using different choices μ. The optimal (in terms of PSNR compared to the reference) is $\mu = 10352$. The image is blurred by a Gaussian blur kernel of size 9×9, $\sigma = 5$. Addition Gaussian noise is added to the image so that the blurred signal to noise ratio (BSNR) is 40dB.



Input $\mu = 0.01$ $\mu = 1$ $\mu^* = 7$ $\mu = 50$

Fig. 2. TV/L1 Image recovery using different choices μ. The optimal (in terms of PSNR compared to the reference) is $\mu = 7$. The image is blurred by a Gaussian blur kernel of size 9×9, $\sigma = 1$. 10% of the pixels are corrupted by salt and pepper noise. Image source: [39].

prior to solving the minimization. Recent advances in the operator splitting methods consider constrained minimization problems [11] so that μ can be replaced by an estimate of the noise level (the noise estimation is performed using a third party algorithm). However, from our experience, it is often easier to choose μ than estimating the noise level, for the noise characteristic of a video is never known exactly. Empirically, a reasonable μ for a natural image (and video sequence) typically lies in the range $[10^3, 10^5]$. Fig. 2 shows the recovery results by using different values of μ. In case of TV/L1 minimization, μ is typically lying in the range $[0.1, 10]$.

2) Choosing ρ_r: One of the major differences between the proposed algorithm and FTVd 4.0 [31] [1] is the update of ρ_r. In [31], ρ_r is a fixed constant. However, as mentioned in [40], the method of multipliers can exhibit a faster rate of convergence by adapting the following parameter update scheme:

$$\rho_r = \begin{cases} \gamma\rho_r, & \text{if } \|\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}\|_2 \geq \alpha\|\mathbf{u}_k - \mathbf{Df}_k\|_2, \\ \rho_r, & \text{otherwise.} \end{cases} \tag{24}$$

Here, the condition $\|\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}\|_2 \geq \alpha_r\|\mathbf{u}_k - \mathbf{Df}_k\|_2$ specifies the constraint violation with respect to a constant α_r. The intuition is that the quadratic penalty $\frac{\rho}{2}\|\mathbf{u} - \mathbf{Df}\|^2$ is a convex surface added to the original objective function $\mu\|\mathbf{Hf} - \mathbf{g}\|^2 + \|\mathbf{u}\|_1$ so that the problem is guaranteed to be strongly convex [29]. Ideally, the residue $\frac{\rho_r}{2}\|\mathbf{u}_k - \mathbf{Df}_k\|^2$ should decrease as k increases. However, if $\frac{\rho_r}{2}\|\mathbf{u}_k - \mathbf{Df}_k\|^2$ is not decreasing for some reasons, one can increase the weight of the penalty $\frac{\rho_r}{2}\|\mathbf{u} - \mathbf{Df}\|^2$ relative to the objective so that $\frac{\rho_r}{2}\|\mathbf{u} - \mathbf{Df}\|^2$ is forced to be reduced. Therefore, given α and γ where $0 < \alpha < 1$ and $\gamma > 1$, Equation (24) makes sure that the constraint violation is decreasing asymptotically. In the steady state as $k \to \infty$, ρ_r becomes a constant [41]. The update for ρ_o in TV/L1 follows a similar approach.

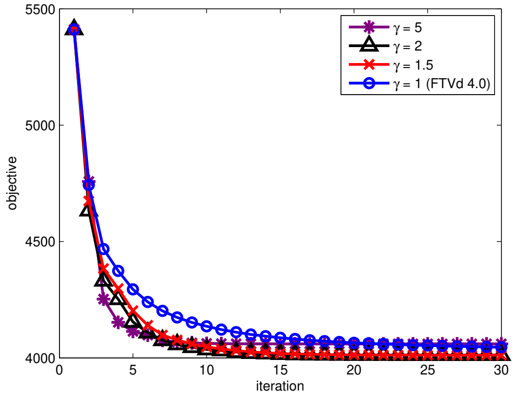


Fig. 3. Convergence profile of the proposed algorithm for deblurring the image “cameraman.tif”. The four colored curves show the rate of convergence using different values of γ, where γ is the multiplication factor for updating ρ_r.

The initial value of ρ_r is chosen to be within the range of $[2, 10]$. This value cannot be large (in the order of 100), because the role of the quadratic surface $\|\mathbf{u} - \mathbf{Df}\|^2$ is to perturb the original objective function so that it becomes strongly convex. If the initial value of ρ_r is too large, the solution of the original problem may not be found. However, ρ_r cannot be too small either, for otherwise the effect of the quadratic surface $\|\mathbf{u} - \mathbf{Df}\|^2$ becomes negligible. Empirically, we find that $\rho_r = 2$ is robust to most restoration problems.

D. Convergence

Fig. 3 illustrates the convergence profile of the TV/L2 algorithm in a typical image recovery problem. In this test, the image “cameraman.tif” (size 256×256, gray-scaled) is blurred by a Gaussian blur kernel of size 9×9 and $\sigma = 1$. Gaussian noise is added so that the blurred signal to noise

[1] The most significant difference is that FTVd 4.0 supports only images whereas the proposed algorithm supports videos.

TABLE I

SENSITIVITY ANALYSIS OF PARAMETERS. MAXIMUM AND MINIMUM PSNR (dB) FOR A RANGE OF ρ_r, γ AND α. IF A PARAMETER IS NOT THE VARIABLE, IT IS FIXED AT THE DEFAULT VALUES: $\rho_r = 2$, $\gamma = 2$, $\alpha = 0.7$.

Image no.	$1.5 \leq \rho_r \leq 10$			$1 \leq \gamma \leq 5$			$0.5 \leq \alpha \leq 0.9$		
	Max	Min	Difference	Max	Min	Difference	Max	Min	Difference
1	28.6468	28.8188	0.1719	28.6271	28.7931	0.1661	28.5860	28.8461	0.2601
2	31.3301	31.4858	0.1556	31.7720	32.0908	0.3188	31.0785	31.5004	0.4219
3	31.7009	31.9253	0.2244	31.9872	32.0847	0.0976	31.7238	31.9833	0.2596
4	33.6080	33.8427	0.2346	33.9994	34.0444	0.0450	34.1944	34.6197	0.4252
5	36.2843	36.5184	0.2341	36.1729	36.3173	0.1444	35.9405	36.7737	0.8332
6	32.0193	32.3859	0.3666	32.2805	32.4795	0.1990	31.9998	32.4207	0.4208
7	29.2861	29.7968	0.5107	29.5890	29.7408	0.1518	29.8872	30.1685	0.2813
8	30.0598	30.4347	0.3749	29.6344	29.9748	0.3404	29.4519	29.7627	0.3108
9	34.4951	34.7675	0.2724	34.5234	34.7378	0.2144	34.3567	34.9726	0.6159
10	29.5555	30.1231	0.5676	29.3502	29.5715	0.2213	29.4009	29.6558	0.2549
11	28.6291	29.1908	0.5617	28.6711	28.9846	0.3135	28.7760	29.0099	0.2340
12	31.6657	31.7473	0.0815	31.2254	31.3172	0.0918	31.3596	31.5423	0.1827
13	35.5306	35.9015	0.3710	35.4584	35.7442	0.2858	36.0163	36.2163	0.2000
14	36.8008	36.9204	0.1196	37.1039	37.1956	0.0917	36.6822	37.1470	0.4648
15	32.0469	32.0969	0.0501	32.4076	32.5918	0.1843	32.0101	32.5421	0.5320
16	31.5836	31.6572	0.0736	31.5975	31.9582	0.3607	31.3778	31.6027	0.2249
17	32.2500	32.6248	0.3748	32.8744	33.0967	0.2223	32.5141	32.8665	0.3524
18	32.6311	33.0377	0.4066	32.2999	32.5472	0.2473	32.9494	33.1908	0.2414
19	28.4927	29.1870	0.6943	28.6654	28.8488	0.1834	28.7902	29.0220	0.2318
20	30.2615	30.6387	0.3771	30.3235	30.6007	0.2772	30.3351	30.7206	0.3855

TABLE II

COMPARISONS BETWEEN OPERATOR SPLITTING METHODS FOR TV/L2 MINIMIZATION

	Fast-TV (2008) [26]	FTVd 3.0 (2008) [10]	FTVd 4.0 (2009) [31] Split Bregman (2008) [42] Constrained TV (2010) [11]	Proposed
Principle	Half quadratic penalty	Half quadratic penalty	Operator Splitting	Operator Splitting
Domain	Gray-scale image	Gray-scale image Color image	Gray-scale image Color image	Gray-scale image Color image Video
Regularization	Spatial TV	Spatial TV	Spatial TV	Spatial-Temporal TV
Penalty Parameter	$\rho_r \to \infty$	$\rho_r \to \infty$	constant ρ_r	Update ρ_r based on constraint violation
Speed [3]	83.39 sec	7.86 sec	2.94 sec	1.79 sec

ratio (BSNR) is 40dB. To visualize the effects of the parameter update scheme, we set the initial value of ρ_r to be $\rho_r = 2$, and let $\alpha = 0.7$. Referring to Equation (24), ρ_r is increased by a factor of γ if the condition is satisfied. Note that [31](FTVd 4.0) is a special case when $\gamma = 1$, whereas the proposed algorithm allows the user to vary γ.

In Fig. 3, the y-axis is the objective value $\frac{\mu}{2}\|\mathbf{H}\mathbf{f}_k - \mathbf{g}\|^2 + \|\mathbf{f}_k\|_{TV2}$ for the k-th iteration, and the x-axis is the iteration number k. As shown in the figure, an appropriate choice of γ improves the rate of convergence significantly. However, if γ is too large, the algorithm is not converging to the solution. Empirically, we find that $\gamma = 2$ is robust to most of the image and video problems.

E. Sensitivity Analysis

Table I illustrates the sensitivity of the algorithm to the parameters ρ_r, γ and α. In this test, twenty images are blurred by a Gaussian blur kernel of size 9×9 with variance $\sigma = 1$. The blurred signal to noise ratio (BSNR) is 30dB. For each image, two of the three parameters (ρ_r, γ and α) are fixed at their default values $\rho_r = 2$, $\gamma = 2$, $\alpha = 0.7$, whereas one of them are varying within the range specified in Table I. The stopping criteria of the algorithm is $\|\mathbf{f}_{k+1} - \mathbf{f}_k\|_2 / \|\mathbf{f}_k\| \leq 10^{-3}$, $\mu = 10000$ and $(\beta_x, \beta_y, \beta_t) = (1, 1, 0)$ for all images. The maximum PSNR, minimum PSNR and the difference are reported in Table I. Referring to the values, it can be calculated that the average maximum-to-minimum PSNR differences among all twenty images for ρ_r, γ and α are 0.311dB, 0.208dB and 0.357dB respectively. For an average PSNR difference in the order of 0.3dB, the perceivable difference is small [2].

F. Comparison with Existing Operator Splitting Methods

The proposed algorithm belongs to the class of operator splitting methods. Table II summarizes the differences between the proposed method and some existing methods [3].

IV. APPLICATIONS

In this section we demonstrate three applications of the proposed algorithm, namely (1) video deblurring, (2) video disparity refinement, and (3) restore videos distorted

[2] It should be noted that the optimization problem is identical for all parameter settings. Therefore, the correlation between the PSNR and visual quality is high.

[3] The speed comparison is based on deblurring "lena.bmp" (512×512, gray scaled), which is blurred by a Gaussian blur kernel of size 9×9, $\sigma = 5$, BSNR = 40dB. The machine used is Intel Qual Core 2.8GHz, 4GB RAM, Windows 7/ MATLAB 2010. Comparisons between FTVd 4.0 and the proposed method are based on $\rho_r = 2$. If $\rho_r = 10$ (default setting of FTVd 4.0), then the run time are 1.56 sec and 1.28 sec for FTVd 4.0 and the proposed method, respectively.

by hot-air turbulence. Due to limited space, more results are available at `http://videoprocessing.ucsd.edu/~stanleychan/deconvtv`.

A. Video Deblurring

1) Spatially Invariant Blur: We first consider the class of spatially invariant blur. In this problem, the t-th observed image $g(x, y, t)$ is related to the true image $f(x, y, t)$ as

$$g(x, y, t) = h(x, y) * f(x, y, t) + \eta(x, y, t).$$

Note that the spatially invariant blur kernel $h(x, y)$ is assumed to be identical for all time t.

The typical method to solve a spatially invariant blur is to consider the model

$$\mathbf{g}_k = \mathbf{H}\mathbf{f}_k + \eta,$$

and apply a frame-by-frame approach to recover $\mathbf{f}_k$ individually. In [21], the authors considered the following minimization

$$\underset{\mathbf{f}_k}{\text{minimize}}\ \|\mathbf{H}\mathbf{f}_k - \mathbf{g}_k\|^2 + \lambda_S \sum_i \|\mathbf{D}_i \mathbf{f}_k\|_1 + \lambda_T \|\mathbf{f}_k - \mathbf{M}_k \hat{\mathbf{f}}_{k-1}\|^2,$$

where $\hat{\mathbf{f}}_{k-1}$ is the solution of the $k-1$-th frame and $\mathbf{M}_k$ is the motion compensation operator that maps the coordinates of $\mathbf{f}_{k-1}$ to the coordinates of $\mathbf{f}_k$. The operators $\mathbf{D}_i$ are the spatial forward finite difference operators oriented at angles $0\,°$, 45°, 90° and 135°. The regularization parameters λ_S and λ_T control the relative emphasis put on the spatial and temporal smoothness.

Another method to solve the spatially invariant blur problem is to apply the multichannel approach by modeling the imaging process as [19], [20]

$$\mathbf{g}_i = \mathbf{H}\mathbf{M}_{i,k}\mathbf{f}_k + \eta,$$

for $i = k - m, \ldots, k, \ldots, k + m$, where m is the size of the temporal window (typically ranged from 1 to 3). $\mathbf{M}_{i,k}$ is the motion compensation operator that maps the coordinates of $\mathbf{f}_k$ to the coordinates of $\mathbf{g}_i$. The k-th frame can be recovered by solving the following minimization [19]

$$\underset{\mathbf{f}_k}{\text{minimize}} \sum_{i=k-m}^{k+m} a_i \|\mathbf{H}\mathbf{M}_{i,k}\mathbf{f}_k - \mathbf{g}_i\|^2 + \lambda \|\mathbf{f}_k\|_{TV2}, \tag{25}$$

where a_i is a constant and $\|\mathbf{f}_k\|_{TV2}$ is the isotropic total variation on the k-th frame. The method presented in [20] replaces the objective function by a weighted least-squares and the isotropic total variation regularization function by a weighted total variation. The weights are adaptively updated in each iteration.

A drawback of these methods is that the image recovery result depends heavily on the accuracy of motion estimation and compensation. Especially in occlusion areas, the assumption that $\mathbf{M}_{i,k}$ is a one-to-one mapping [43] fails to hold. Thus, $\mathbf{M}_{i,k}$ is not a full rank matrix and $\mathbf{M}_{i,k}^T \mathbf{M}_{i,k} \neq \mathbf{I}$. As a result, minimizing $\|\mathbf{H}\mathbf{M}_{i,k}\mathbf{f}_k - \mathbf{g}_i\|^2$ can lead to serious error. There are methods to reduce the error caused by rank deficiency of $\mathbf{M}_{i,k}$, for example the concept of *unobservable pixel* introduced in [19], but the restoration result depends on the effectiveness of how the unobservable pixels are selected.

Another drawback of these methods is the computation time. For spatially invariant blur, the blur operator $\mathbf{H}$ is a block circulant matrix. However, in the multichannel model, the operator $\mathbf{H}\mathbf{M}_{i,k}$ is not a block circulant matrix. The block-circulant property is a critical factor to speed as it allows Fourier Transform methods. For methods in [19], [20], conjugate gradient (CG)is used to solve the minimization task. While the total number of CG iterations may be few, the per iteration run time can be long.

Table III illustrates the differences between various video restoration methods.

TABLE IV
PSNR, E_S AND E_T VALUES FOR FOUR VIDEO SEQUENCES BLURRED BY GAUSSIAN BLUR KERNEL 9×9, $\sigma = 1$, $BSNR = 30dB$.

		"Foreman"	"Salesman"	"Mother"	"News"
PSNR (dB)	Blurred	28.6197	29.9176	32.5705	28.1106
	[24]	31.6675	33.0171	36.1493	34.0113
	[19]	32.5500	33.8408	38.2164	34.1207
	[21]	33.2154	33.8618	39.6991	34.7133
	Proposed	**33.7864**	**34.7368**	**40.0745**	**35.8813**
E_S $(\times 10^4)$	[24]	1.2067	1.1706	0.82665	1.3764
	[19]	1.1018	1.0743	0.71751	1.2146
	[21]	**1.0076**	**0.9934**	0.61544	1.123
	Proposed	1.0930	1.0105	**0.61412**	**1.1001**
E_T $(\times 10^3)$	[24]	10.954	3.3195	3.7494	4.6484
	[19]	10.827	2.4168	2.9397	3.7503
	[21]	10.202	2.5471	2.7793	3.3623
	Proposed	**9.3400**	**1.9948**	**2.0511**	**2.6165**

Our approach to solve spatially invariant blur problem shares the same insight as [24] which does *not* consider motion compensation. The temporal error is handled by the spatio-temporal total variation $\|\mathbf{D}\mathbf{f}\|_2 = \sum_i \sqrt{|[\mathbf{D}_x\mathbf{f}]_i|^2 + |[\mathbf{D}_y\mathbf{f}]_i|^2 + |[\mathbf{D}_t\mathbf{f}]_i|^2}$. An intuition to this approach is that the temporal difference $\mathbf{f}_k - \mathbf{f}_{k-1}$ can be classified as temporal *edge* and temporal *noise*. The temporal edge is the intensity change caused by object movements, whereas the temporal noise is the artifact generated in the minimization process. Similar to the spatial total variation, the temporal total variation preserves the temporal edges while reducing the temporal noise. Moreover, the space-time volume preserves the block circulant structure of the operator, thus leading to significantly faster computation.

Table IV and Fig. 4 show the comparisons between [24], [19], [21] and the proposed method on spatially invariant blur. The four testing video sequences are blurred by a Gaussian blur kernel of size 9×9 with $\sigma = 1$. Additive Gaussian noise are added so that the blurred signal to noise ratio (BSNR) is 30dB.

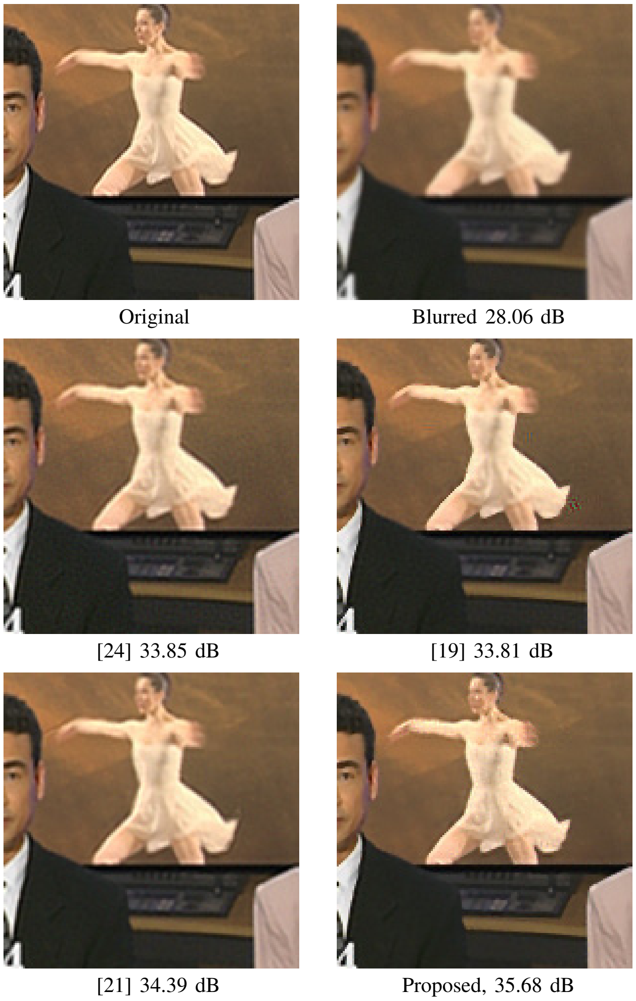
The specific settings of the methods are as follows. For [24], we consider the minimization

$$\underset{\mathbf{f}}{\text{minimize}}\ \mu\|\mathbf{H}\mathbf{f} - \mathbf{g}\|^2 + \beta_x^2\|\mathbf{D}_x\mathbf{f}\|^2 + \beta_y^2\|\mathbf{D}_y\mathbf{f}\|^2 + \beta_t^2\|\mathbf{D}_t\mathbf{f}\|^2$$

and set the parameters empirically for the best recovery quality: $\mu = 200$, $(\beta_x, \beta_y, \beta_t) = (1,\,1,\,1.25)$. For [19], instead of using the CG presented in the paper, we use a modification of the proposed augmented Lagrangian method to speed up the computation. Specifically, in solving the $\mathbf{f}$-subproblem we used conjugate gradient (LSQR [44]) to accommodate the non-block-circulant operator $\mathbf{H}\mathbf{M}_{i,k}$. The motion estimation is

TABLE III
COMPARISONS BETWEEN VIDEO RESTORATION METHODS

	Belekos 2010 [20]	Ng 2007 [19]	Chan 2011 [21]	Shechtman 2005 [24]	Proposed
Class of problem	super-resolution	super-resolution	deblurring	super-resolution	deblurring
Approach	frame-by-frame	frame-by-frame	frame-by-frame	space-time volume	space-time volume
Spatial Consistency	$\sum_i w_i \sqrt{[\mathbf{D}_x\mathbf{f}]_i^2 + [\mathbf{D}_y\mathbf{f}]_i^2}$	$\sum_i \sqrt{[\mathbf{D}_x\mathbf{f}]_i^2 + [\mathbf{D}_y\mathbf{f}]_i^2}$	$\sum_i \|\mathbf{D}_i\mathbf{f}\|_1$	$\|\mathbf{D}_x\mathbf{f}\|^2 + \|\mathbf{D}_y\mathbf{f}\|^2$	$\|\mathbf{f}\|_{TV2}$, Equation (5)
Temporal Consistency	weighted $\|\mathbf{HM}_{i,k}\mathbf{f}_k - \mathbf{g}_i\|^2$	$\|\mathbf{HM}_{i,k}\mathbf{f}_k - \mathbf{g}_i\|^2$	$\|\mathbf{f}_k - \mathbf{M}_k\hat{\mathbf{f}}_{k-1}\|^2$	$\|\mathbf{D}_t\mathbf{f}\|^2$	$\|\mathbf{f}\|_{TV2}$, Equation (5)
Motion Compensation	Required	Required	Required	Not Required	Not Required
Handle of Motion Blur	spatially variant operator	spatially variant operator	spatially variant operator	3D-FFT	3D-FFT
Objective Function	weighted least-squares	TV/L2	TV/L2 + quadratic penalty	Tikhonov	TV/L2 or TV/L1
Solver	Conjugate gradient	Conjugate gradient	Sub-gradient Projection	Closed-form	Closed-form + Shrinkage


Original

Blurred 28.06 dB

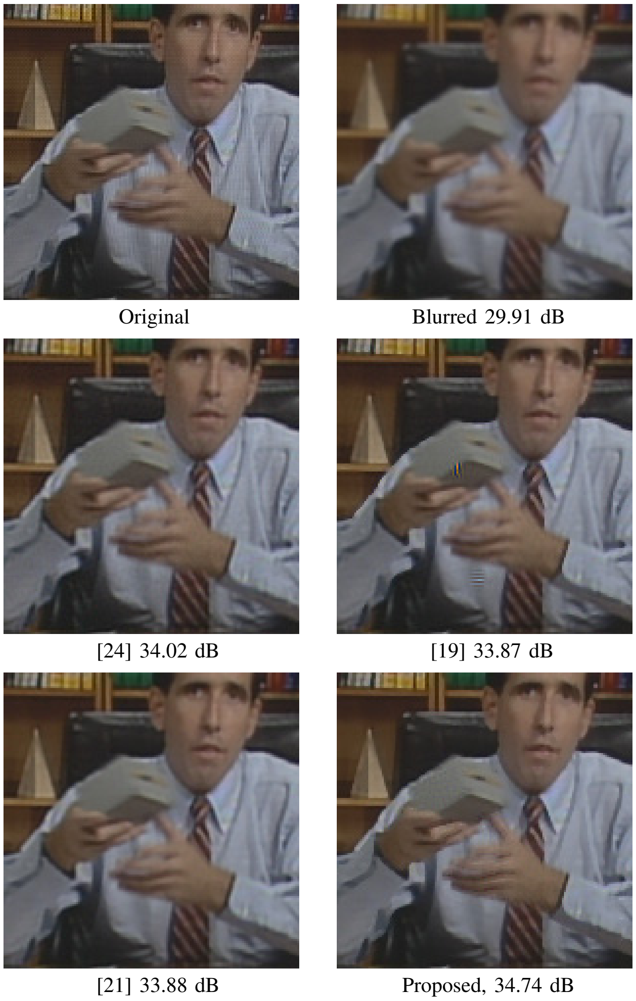
[24] 33.85 dB

[19] 33.81 dB

[21] 34.39 dB

Proposed, 35.68 dB

Fig. 4. "News" sequence, frame no. 100. (a) Original image (cropped for better visualization). (b) Blurred by a Gaussian blur kernel of size 9×9, $\sigma = 1$, BSNR = 30dB. (c)-(f) Results by various methods. (Table IV).


Original

Blurred 29.91 dB

[24] 34.02 dB

[19] 33.87 dB

[21] 33.88 dB

Proposed, 34.74 dB

Fig. 5. "Salesman" sequence, frame no. 10. (a) Original image (cropped for better visualization). (b) Blurred by a Gaussian blur kernel of size 9×9, $\sigma = 1$, BSNR = 30dB. (c)-(f) Results by various methods. (Table IV).

performed using the benchmark full search (exhaustive search) with 0.5 pixel accuracy. The block size is 8×8 and the search range is 16×16. Motion compensation is performed by coordinate transform according to the motion vectors (bilinear interpolation for half pixels). The threshold for unobservable pixels [19] is set as 6 (out of 255), and the regularization parameter is $\lambda = 0.001$ (See Equation (25)). We use the previous and the next frame for the model, i.e. $m = 1$ and let $(a_{k-1}, a_k, a_{k+1}) = (0.5,\ 1,\ 0.5)$ (Using $(1,\ 1,\ 1)$ tends to give worse results). For [21], the regularization parameters are also chosen empirically for the best recovery quality: $\lambda_S = 0.001$ and $\lambda_T = 0.05$.

To compare to these methods, we apply TV/L2 (Algorithm 1) with the following parameters (same for all four videos): $\mu = 2000$, $(\beta_x, \beta_y, \beta_t) = (1,\ 1,\ 1)$. All other parameters take the default setting: $\alpha = 0.7$, $\gamma = 2$, $\rho_r = 2$. The algorithm terminates if $\|\mathbf{f}_k - \mathbf{f}_{k-1}\|/\|\mathbf{f}_{k-1}\| \leq 10^{-3}$.

In Table IV, three quantities are used to evaluate the

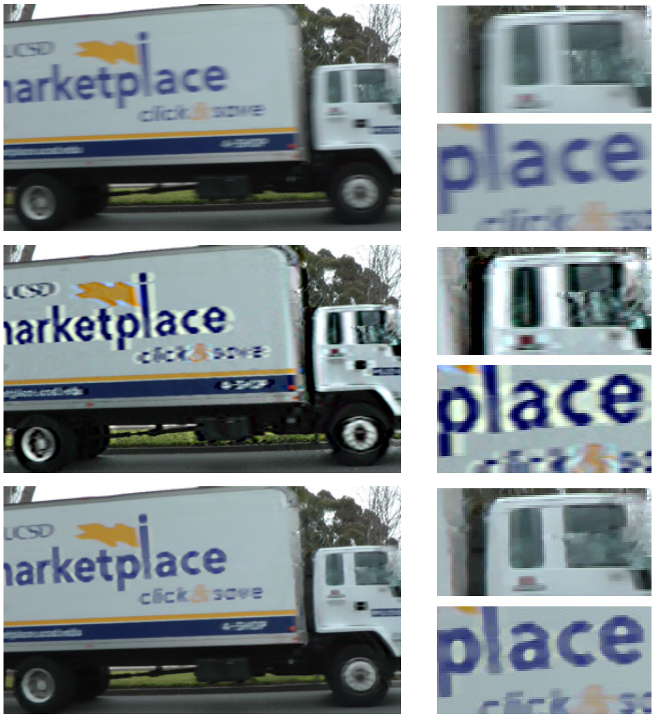


Fig. 6. “Market Place” sequence, frame no. 146. Top: The original observed video sequences. Middle: Result of [24]. Bottom: Result of the proposed method.

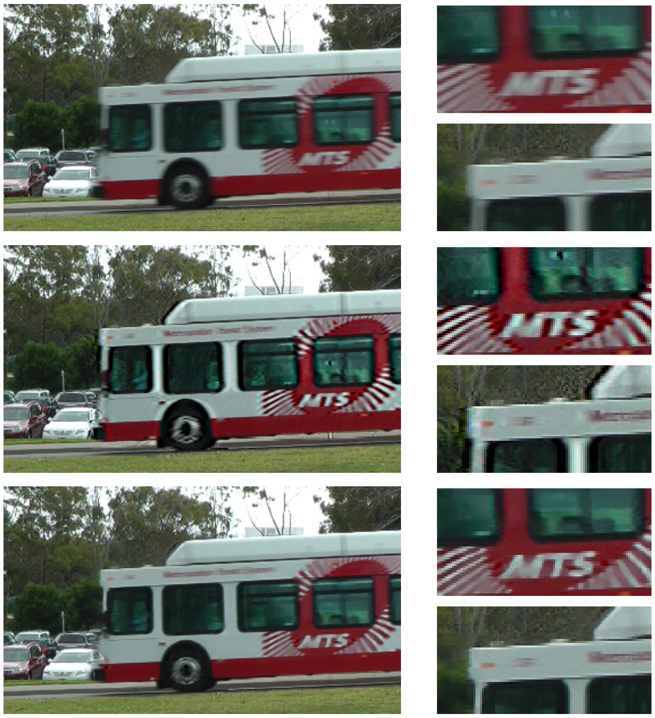


Fig. 7. “Super Loop” sequence, frame no. 28. Top: The original observed video sequences. Middle: Result of [24]. Bottom: Result of the proposed method.

performance of the algorithms. Peak signal to noise ratio (PSNR) measures the image fidelity. The spatial total variation E_S is defined as $E_S = \sum_i \sqrt{|[\mathbf{D}_x\mathbf{f}]_i|^2 + |[\mathbf{D}_y\mathbf{f}]_i|^2}$ for each frame and the temporal total variation E_T is defined as $E_T = \sum_i |[\mathbf{D}_t\mathbf{f}]_i|$ for each frame [21]. The average (over all frames) PSNR, E_S and E_T are listed in Table IV.

Referring to the results, it can be seen that the proposed algorithm produces the highest PSNR values while keeping E_S and E_T at a low level. It is worth noting that [24] is equivalent to the three-dimensional Wiener deconvolution (regularized). Therefore, there exists a closed form solution but the result looks more blurry than the other methods. Among the four methods, both [19] and [21] use motion estimation and compensation. However, [19] is more sensitive to the motion estimation error - motion estimation error in some fast moving areas are amplified in the deblurring step. [21] is more robust to motion estimation error, but the computation time is significantly longer than the proposed method. The run time of [19] and [21] are approximately 100 seconds per frame (per color channel) whereas the proposed algorithm only requires approximately 2 seconds per frame (per color channel). These statistics are based on recovering videos of size 288 × 352, using a PC with Intel Qual Core 2.8 GHz, 4GB RAM, Windows 7/ MATLAB 2010.

2) Spatially Variant Motion Blur: The proposed algorithm can be used to remove spatially-variant motion blur. However, since motion blurred videos often have low temporal resolution, frame rate up conversion algorithms are needed to first increase the temporal resolution before applying the proposed method (See [24] for detailed explanations). To this end, we apply [45] to upsample the video by a factor of 8. Consequently, the motion blur kernel can be modeled as

$$h(x,y,t) = \begin{cases} 1/T, & \text{if } x = y = 0, \text{ and } 0 \le t \le T, \\ 0, & \text{otherwise,} \end{cases}$$

where $T = 8$ in this case.

Fig. 6 shows frame no. 146 of the video sequence “Market Place”, and Fig. 7 shows frame no. 28 of the video sequence “Super Loop”. The videos are captured by a Panasonic TM-700 video recorder with resolution 1920×1080p at 60 fps. For computational speed we down sampled the spatial resolution by a factor of 4 (so the resolution is 480×270). The parameters of the proposed algorithm are chosen empirically as $\mu = 1000$, $(\beta_x, \beta_y, \beta_t) = [1, 1, 5]$. There are not many relevant video motion deblurring algorithms for comparison (or unavailable to be tested). Therefore, we are only able to show the results of [24], using parameters $\mu = 1000$, $(\beta_x, \beta_y, \beta_t) = [1, 1, 2.5]$.

As shown in Fig. 6 and Fig. 7, the proposed algorithm produces a significantly higher quality result than [24]. We also tested a range of parameters μ and β’s for [24]. However, we observe that the results are either over-sharpened (serious ringing artifacts), or under-sharpened (not enough deblurring).

3) Limitation: The proposed algorithm requires considerably less memory than other total variation minimization algorithms such as interior point methods. However, for high definition (HD) videos, the proposed algorithm still has memory issue as the size of the space-time volume is large. While one can use fewer frames to lower the memory demand, trade off in the recovery quality should be expected.

Another bottleneck of the proposed algorithm is the sensitivity to the frame-rate conversion algorithm. At object boundaries where the motion estimation algorithm fails to provide accurate estimates, the estimation error in the deblurring step will be amplified. This happens typically to areas with non-uniform and rapid motion.

B. *Video Disparity Refinement*

1) Problem Description: Our second example is disparity map refinement. Disparity is proportional to the reciprocal of the distance between the camera and the object (i.e., depth). Disparity maps are useful for many stereo video processing applications, including object detection in three-dimensional space, saliency for stereo videos, stereo coding and view synthesis etc.

There are numerous papers on generating one disparity map based on a pair of stereo images [46]. However, all of these methods cannot be extended to videos because the energy functions are considered in a frame-by-frame basis. Although there are works in enforcing temporal consistency for adjacent frames, such as [47] and [48], the computational complexity is high.

We propose to estimate the video disparity in two steps. In the first step, we combine the locally adaptive support weight [49] and the dual cross bilateral grid [50] to generate an initial disparity estimate. Since this method is a frame-by-frame method, spatial and temporal consistency is poor. In the second step, we consider the initial video disparity as a space-time volume and solve the TV/L1 minimization problem

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

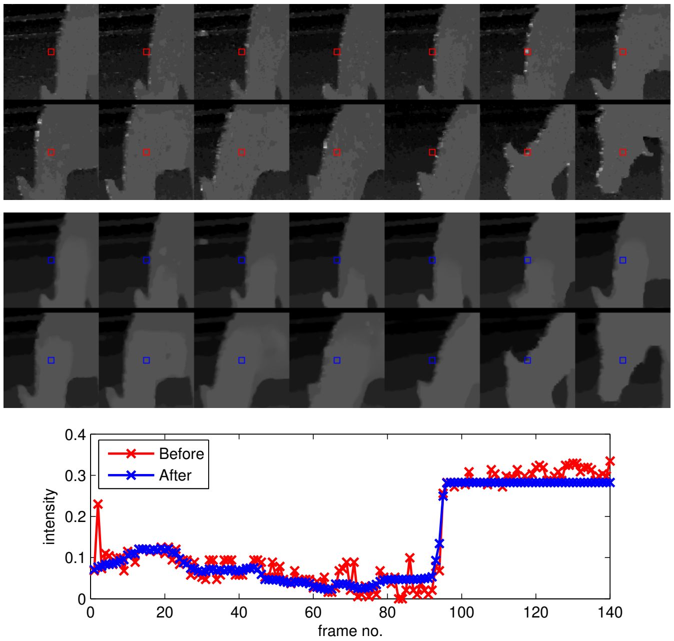
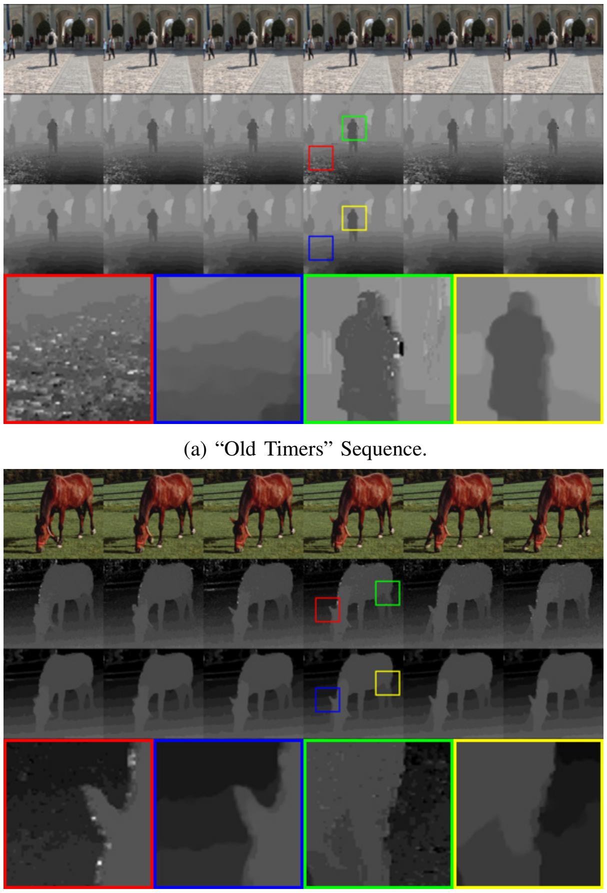


Fig. 8. Top: Before applying the proposed TV/L1 algorithm; Middle: After applying the proposed TV/L1 algorithm. Bottom: Trace of a pixel along the time axis.

There are two reasons for choosing TV/L1 instead of TV/L2 in refining video disparity. First, disparity is a piece-wise constant function with quantized levels, and across the flat regions there are sharp edges. As shown in Fig. 8 (bottom), the estimation error behaves like outliers in a smooth function. Therefore, to reduce the estimation error, one can consider a robust curve fitting as it preserves the shape of the data while suppressing the outliers.

The second reason for using TV/L1 is that the one-norm $\|\mathbf{f}-\mathbf{g}\|_1$ is related to the notion of percentage of bad pixels, a quantity commonly used to evaluate disparity estimation algorithms [46]. Given a ground truth disparity $\mathbf{f}^*$, the number of bad pixels of an estimated disparity $\mathbf{f}$ is the cardinality of the set $\{i| \;\; |[\mathbf{f}-\mathbf{f}^*]_i| > \tau\}$ for some threshold τ. In the absence of ground truth, the same idea can be used with a reference disparity (e.g., $\mathbf{g}$). In this case, the cardinality of the set $\Omega_\tau = \{i| \;\; |[\mathbf{f}-\mathbf{g}]_i| > \tau\}$, denoted by $|\Omega_\tau|$, is the number of bad pixels of $\mathbf{f}$ with respect to (w.r.t) $\mathbf{g}$. Therefore, minimizing $|\Omega_\tau|$ is equivalent to minimizing the number of bad pixels of $\mathbf{f}$ w.r.t. $\mathbf{g}$. However, this problem is non-convex and is NP-hard. In order to alleviate the computational difficulty, we set $\tau = 0$ so that $|\Omega_\tau| = \|\mathbf{f}-\mathbf{g}\|_0$, and convexify $\|\mathbf{f}-\mathbf{g}\|_0$ by $\|\mathbf{f}-\mathbf{g}\|_1$. Therefore, $\|\mathbf{f}-\mathbf{g}\|_1$ can be regarded as the convexification of the notion of percentage bad pixels.



(a) “Old Timers” Sequence.

(b) “Horse” Sequence.

Fig. 9. Video disparity estimation. First row: Left view of the stereo video. Second row: Initial disparity estimate. Third row: Refinement using the proposed method with parameters $\mu = 0.75$, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$, $\alpha = 0.7$, $\rho_r = 2$, $\rho_o = 100$, $\gamma = 2$. Last row: Zoom-in comparisons.

2) Video Results: Two real videos (“Horse” and “Old Timers”) are tested for the proposed algorithm. These stereo

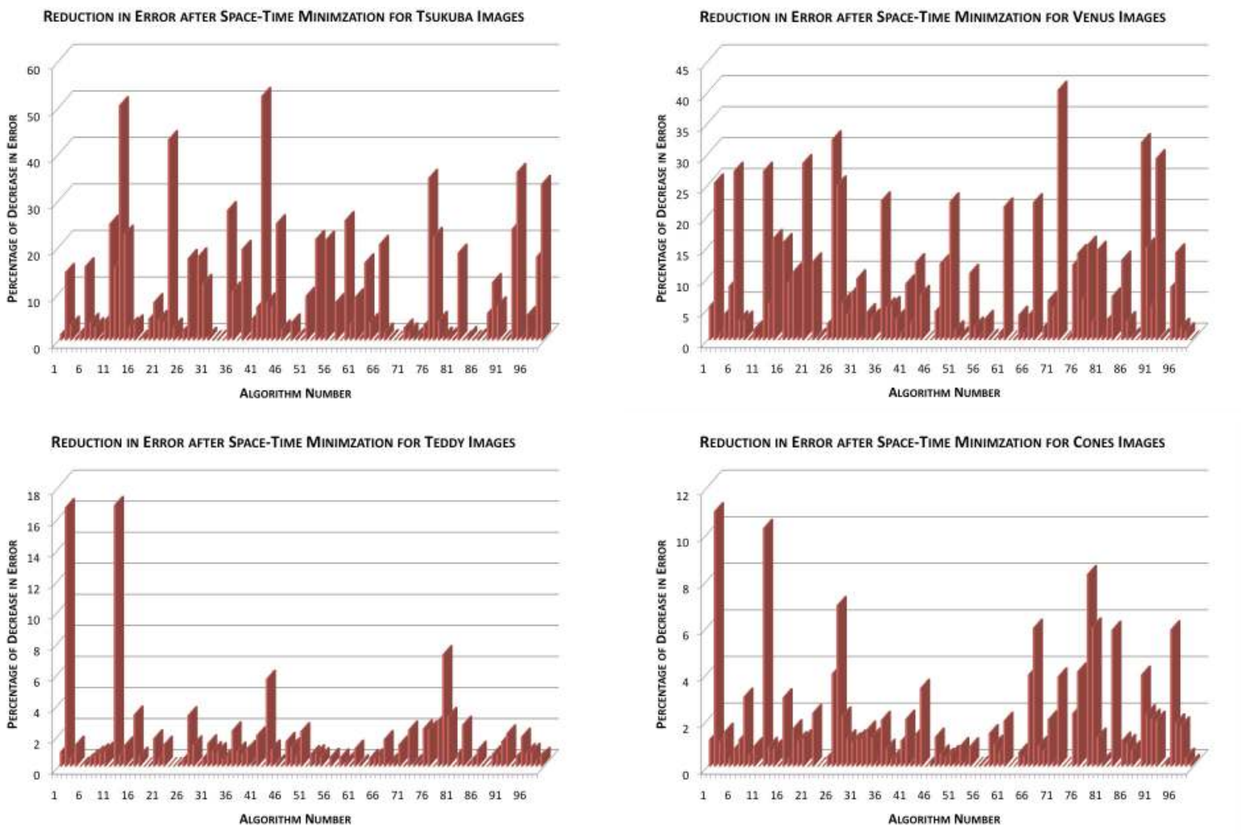
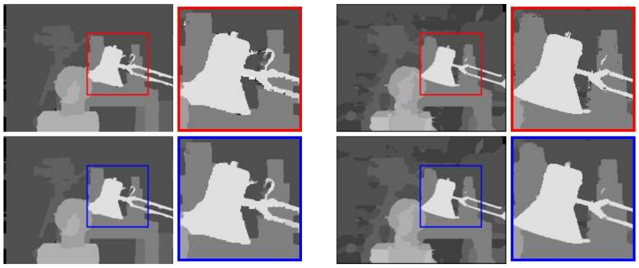


Fig. 10. Percentage error reduction (in terms of number of bad pixels) by applying the proposed algorithm to all 99 methods on the Middlebury stereo database.

videos are downloaded from http://sp.cs.tut.fi/mobile3dtv/stereo-video/. Fig. 9 illustrates the results. The first row of Fig. 9 shows the left view of the stereo video. The second row shows the results of applying [49], [50] to the stereo video. Note that we are implementing a spatio-temporal version of [50], which uses adjacent frames to enhance the temporal consistency. However, the estimated disparity is still noisy, especially around the object boundaries. The third row shows the result of applying the proposed TV/L1 minimization to the initial disparity estimated in the second row. It should be noted that the proposed TV/L1 minimization improves not only the flat interior region, but also the object boundary (e.g. the arm of the man in "Old Timers" sequence), an area that [49], [50] are unable to handle.

3) Image Results: The effectiveness of the proposed algorithm can further be elaborated by comparing to the 99 benchmark methods on Middlebury stereo evaluation website [46]. For all 99 methods on Middlebury stereo evaluation website, we download their results and apply the proposed algorithm to improve the spatial smoothness. Note that the proposed algorithm is readily for this test because an image is a single frame video. In this case, we set $(\beta_x, \beta_y, \beta_t) = (1,\ 1,\ 0)$. Fig. 11 shows two of the 99 results (randomly chosen) for the dataset "Tsukuba", and Fig. 10 shows the percentage of error reduction (in terms of number of bad pixels, with threshold 1) by applying the proposed algorithm to all methods on the Middlebury database. The higher bars in the plots indicate that the proposed algorithm reduces the error by a greater amount. It can be observed that the errors are typically reduced by a large margin of over 10%. While there is less error reduction for some datasets, it is important to note that error reduction is *always non-negative*. In other words, the proposed algorithm always improves the initial disparity estimate. Furthermore, for *every* single algorithm, we provide improvement in at least one of the image sets.



Algorithm no. 8 Algorithm no. 78

Fig. 11. Image disparity refinement on algorithms no. 8 and 78 (randomly chosen) from Middlebury for "Tsukuba". Red box: Before applying the proposed method; Blue box: After applying the proposed method. $\mu \in [0.1,\ 1]$ is found exhaustively with increment 0.1, $(\beta_x, \beta_y, \beta_t) = (1, 1, 0)$, $\alpha = 0.7$, $\rho_r = 2$, $\rho_o = 100$, $\gamma = 2$.

4) Limitations: A limitation of the proposed algorithm is that it is unable to handle large and consistent error results from poor initial disparity estimation algorithm. This happens especially in large occlusion areas, repeating texture regions, or frames consisting of rapid motions. We are currently seeking methods to feedback the TV/L1 result to the initial disparity estimation so that the algorithm are more robust to these errors.

C. Videos Distorted by Hot-Air Turbulence

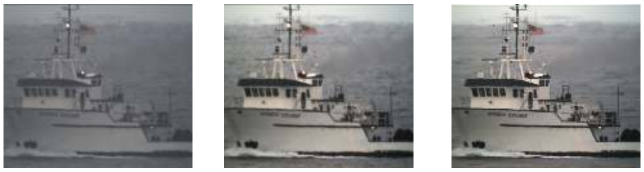
1) Problem Description: Our third example is the stabilization of videos distorted by hot-air turbulence effects. In the presence of hot-air turbulence, the refractive index along the transmission path of the light ray is spatially and temporally varying [51]. Consequently, the path differences and hence the phases of the light rays are also spatially and temporally varying. As a result, the observed image is distorted by geometric warping, motion blur and sometimes out-of-focus blur. This type of distortion is generally known as the hot-air turbulence effect.

There are various methods to overcome imaging through hot-air turbulence. For example, the speckle imaging technique [51] assumes that the refractive index is changing randomly but is also statistically stationary [52], [53]. Consequently, by averaging enough number of frames, the geometric distortion will be smoothed out. Then a deconvolution algorithm can be used to remove the blur.

The drawback of the speckle imaging technique is that the average operation makes the deblurring process challenging. Therefore, Zhu and Milanfar [54], Shimizu et. al. [55] proposed to first compensate the geometric distortion using non-rigid registration [56], and then deblur the images using deconvolution algorithms. The limitation is that non-rigid registration works well only when the geometric distortion can be adjusted by all the control points in the grid [56]. However, imaging through hot-air turbulence contains both large area distortion (perceived as waving) and small disturbance (perceived as jittering). If non-rigid registration has to be used to compensate small disturbance, then the number of control points will be huge, making the computation not possible. There are other methods such as lucky frame/ region fusion approach [57], [58]. However, these methods cannot handle small disturbance effectively either.

Using the same methodology as we used for video deblurring, we consider the video as a space-time volume and minimize the TV/L2 problem. Our intuition is that the small hot-air turbulence can be regarded as temporal noise whereas the object movement is regarded as temporal edge. Under this framework, spatially invariant blur can also be incorporated. If the input video originally has a low contrast, a preprocessing step using gray level grouping (GLG) [59], [60] can be used (See Fig. 12).

2) Experiments: Fig. 13 shows the snapshots (zoom-in) of a video sequence "Acoustic Explorer". In this example, gray level grouping is applied to the input videos so that contrast is enhanced. Then the proposed algorithm is used to reduce the hot-air turbulence effect. A Gaussian blur kernel is assumed in both examples, where the variance is determined empirically. Comparing the video quality before and after applying the proposed method, fewer jittering like artifacts are observed in the processed videos. While this may not be apparent by viewing the still images, the improvement is significant in the 24fps videos [4].



(a) input (b) step 1 [59], [60] (c) step 2: our method.

Fig. 12. Hot-air turbulence removal for the sequence "Acoustic Explorer" - using the proposed method to reduce the effect of hot-air turbulence. (a) A frame of the original video sequence. (b) Step 1: Apply gray level grouping [59], [60] to the input. (c) Step 2: Apply the proposed method to the results of Step 1.

Fig. 14 shows the comparisons without the contrast enhancement by GLG. Referring to the figures, the proposed algorithm does not only reduce the unstable hot-air turbulence effects, it also improves the blur. The word "Empire State" could not be seen clearly in the input sequence, but becomes sharper in the processed sequence.

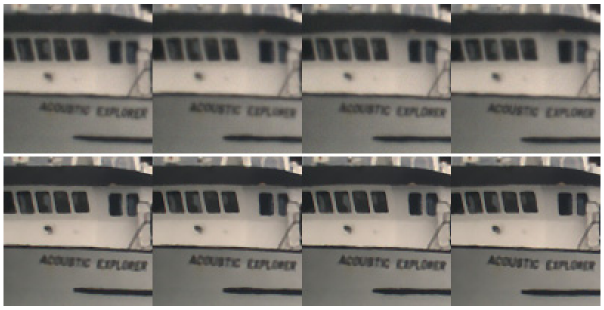


Fig. 13. Zoom-in of "Acoustic Explorer" sequence frame no. 25-28 (object is 2 miles from camera). Top: input video sequence with contrast enhanced by gray level grouping (GLG). Bottom: Processed video by applying the proposed method to the output of GLG.

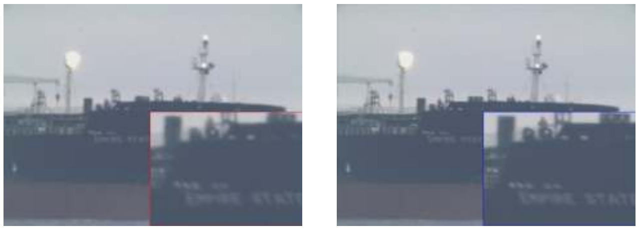


Fig. 14. Snapshot of "Empire State" sequence. Left: input video sequence without GLG. Right: Processed video by applying GLG and the proposed method.

3) Limitation: The experiments above indicates that the proposed algorithm is effective for reducing small hot-air turbulence effects. However, for large area geometric distortions, non-rigid registration is needed. In addition, the general turbulence distortion is spatially and temporally varying, meaning that the point spread function cannot be modeled as one Gaussian function. This issue is an open problem.

[4] Videos are available at http://videoprocessing.ucsd.edu/~stanleychan/deconvtv

V. Conclusion

In this paper, we propose a video deblurring/ denoising algorithm which minimizes a total variation optimization problem for spatial-temporal data. The algorithm transforms the original unconstrained problem to an equivalent constrained problem, and uses an augmented Lagrangian method to solve the constrained problem. With the introduction of spatial and temporal regularization to the spatial-temporal data, the solution of the algorithm is both spatially and temporally consistent.

Applications of the algorithm include video deblurring, disparity refinement and turbulence removal. For video deblurring, the proposed algorithm restores motion-blurred video sequences. The average PSNR is improved, and the spatial and temporal total variation are maintained at an appropriate level, meaning that the restored videos are spatially and temporally consistent. For disparity map refinement, the algorithm removes flickering in the disparity map, and preserves the sharp edges in the disparity map. For turbulence removal, the proposed algorithm stabilizes and deblurs videos taken under the influence of hot air turbulence.

References

[1] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2007.

[2] L. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical Journal*, vol. 79, pp. 745, 1974.

[3] W. Richardson, "Bayesian-based iterative method of image restoration," *JOSA-A*, vol. 62, no. 1, pp. 55–59, 1972.

[4] V. Mesarovic, N. Galatsanos, and A. Katsaggelos, "Regularized constrained total least-squares image restoration," *IEEE Trans. on Image Processing*, vol. 4, pp. 1096–1108, 1995.

[5] P. Hansen, J. Nagy, and D. O' Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundamentals of Algorithms 3. SIAM, 2006.

[6] "FIJI: Fiji Is Just ImageJ," http://pacific.mpi-cbg.de/wiki/index.php/Fiji.

[7] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[8] T. Chan, G. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Scientific Computing*, vol. 20, pp. 1964–1977, 1999.

[9] A. Chambolle, "An algorithm for total variation minimization and applications," *J. of Math. Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[10] 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.

[11] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. on Image Processing*, vol. 19, pp. 2345–2356, September 2010.

[12] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.

[13] B. Lucas, *Generalized image matching by the method of differences*, Ph.D. thesis, Carnegie Mellon University, 1984.

[14] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. on Graphics (SIGGRAPH)*, vol. 27, no. 3, 2008.

[15] S. Dai and Y. Wu, "Motion from blur," in *Proceedings of IEEE CVPR*, 2008.

[16] A. Levin, "Blind motion deblurring using image statistics," in *Proceedings of NIPS*, 2006.

[17] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *Proceedings of IEEE ICCV*, 2007, pp. 1–8.

[18] J. Jia, "Single image motion deblurring using transparency," in *Proceedings of IEEE ICCV*, 2007.

[19] M. Ng, H. Shen, E. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," *EURASIP Journal on Advances in Signal Processing*, p. 74585, 2007.

[20] S. Belekos, N. Galatsanos, and A. Katsaggelos, "Maximum a posteriori video super-resolution using a new multichannel image prior," *IEEE Trans. on Image Processing*, vol. 19, pp. 1451–1464, 2010.

[21] S. Chan and T. Nguyen, "LCD motion blur: modeling, analysis and algorithm," *IEEE Trans. on Image Processing*, 2011, Accepted. Preprint available at http://videoprocessing.ucsd.edu/~stanleychan/.

[22] B. Jähne, *Spatio-temporal image processing: theory and scientific applications*, Springer, 1993.

[23] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. on PAMI*, vol. 29, pp. 1–14, March 2007.

[24] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Trans. on PAMI*, vol. 27, pp. 531–545, 2005.

[25] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-video dynamic super-resolution for grayscale and color sequences," *EURASIP J. Appl. Signal Process.*, pp. 232–232, 2006.

[26] Y. Huang, M. Ng, and Y. Wen, "A fast total variation minimization method for image restoration," *SIAM Multiscale model and simulation*, vol. 7, pp. 774–795, 2008.

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

[28] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. of the Amer. Math. Soc.*, vol. 82, pp. 421–439, 1956.

[29] R. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Math. of Operations Research*, vol. 1, pp. 97–116, 1976.

[30] R. Rockafellar, "Monotone operators and proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, pp. 877, 1976.

[31] M. Tao and J. Yang, "Alternating direction algorithms for total variation deconvolution in image reconstruction," Tech. Rep. TR0918, Nanjing University, China, 2009, available at http://www.optimization-online.org/DB_FILE/2009/11/2463.pdf.

[32] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split Bregman," Tech. Rep. 09-31, ULCA, 2009, available at ftp://ftp.math.ucla.edu/pub/camreport/cam09-31.pdf.

[33] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented lagrangian method for video restoration," in *Proceedings of IEEE ICASSP*, 2011.

[34] R. Khoshabeh, S. Chan, and T. Nguyen, "Spatio-temporal consistency in video disparity estimation," in *Proceedings of IEEE ICASSP*, 2011.

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

[36] G. Golub and C. Van Loan, *Matrix Computation*, Johns Hopkins University Press, second edition, 1989.

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

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

[39] A. Levin, A. Rav-Acha, and D. Lischinski, "Spectral matting," *IEEE Trans. PAMI*, vol. 30, pp. 1699–1712, 2008.

[40] M. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, Fletcher, Ed., pp. 283–298. Academic Press, 1969.

[41] D. Bertsekas, "Multiplier methods: A survey," *Automatica*, vol. 12, pp. 133–145, 1976.

[42] T. Goldstein and S. Osher, "The split bregman algorithm for L1 regularized problems," Tech. Rep. 08-29, UCLA, 2008.

[43] M. Choi, N. Galatsanos, and A. Katsaggelos, "Multichannel regularized iterative restoration of motion compensated image sequences," *J. of Visual Communication and Image Representation*, vol. 7, no. 3, pp. 244–258, 1996.

[44] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. on Mathematical Software*, vol. Vol 8, no. 1, pp. 43–71, March 1982.

[45] Y. Lee and T. Nguyen, "Fast one-pass motion compensated frame interpolation in high-definition video processing," in *Proceedings of IEEE ICIP*, November 2009, pp. 369–372.

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

[47] J. Oh, S. Ma, and C. Kuo, "Disparity estimation and virtual view synthesis from stereo video," in *Proceedings of IEEE ISCAS*, 2007, pp. 993–996.

[48] J. Fan, F. Liu, W. Bao, and H. Xia, “Disparity estimation algorithm for stereo video coding based on edge detecetion,” in *Proceedings of IEEE WCSP*, 2009, pp. 1–5.

[49] K. Yoon and I. Kweon, “Locally adaptive support-weight approach for visual correspondence search.,” in *Proceedings of IEEE CVPR*, 2005.

[50] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. Dodgson., “Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid,” in *Proceedings of ECCV*, 2010.

[51] M. Roggemann and B. Welsh, *Imaging through turbulence*, CRC Press, 1996.

[52] J. Goodman, *Statistical Optics*, Wiley-Interscience, 2000.

[53] J. Goodman, *Introduction to Fourier Optics*, Roberts & Company Publishers, 4 edition, 2004.

[54] X. Zhu and P. Milanfar, “Image reconstruction from videos distorted by atmospheric turbulence,” *Visual Information Processing and Communication*, vol. 7543, no. 1, pp. 75430S, 2010.

[55] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, “Super-resolution from image sequence under influence of hot-air optical turbulence,” in *Proceedings of IEEE CVPR*, 2008.

[56] R. Szeliski and J. Coughlan, “Spline-based image registration,” *International Journal of Computer Vision*, vol. 22(93), pp. 199–218, 1997.

[57] D. Fried, “Probability of getting a lucky short-exposure image through turbulence,” *JOSA*, vol. 68, pp. 1651–1658, 1978.

[58] M. Aubailly, M. Vorontsov, G. Carhat, and M. Valley, “Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach,” in *Proceedings of SPIE*, 2009, vol. 7463.

[59] Z. Chen, B. Abidi, D. Page, and M. Abidi, “Gray-level grouping (GLG): an automatic method for optimized image contrast enhancement-part I,” *IEEE Trans. on Image Processing*, vol. 15, pp. 2290 – 2302, 2006.

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