An Augmented Lagrangian Method for Total Variation Video Restoration

Stanley H. Chan, *Student Member, IEEE*, Ramsin Khoshabeh, *Student Member, IEEE*, Kristofor B. Gibson, *Student Member, 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 is used to iteratively find solutions to 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*—Alternating direction method (ADM), augmented Lagrangian, hot-air turbulence, total variation (TV), video deblurring, video disparity, video restoration.**

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 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 $\mathbf{g}$.

Manuscript received October 29, 2010; revised February 24, 2011 and April 26, 2011; accepted May 05, 2011. Date of publication May 31, 2011; date of current version October 19, 2011. The work of S. Chan was supported by Croucher Foundation Scholarship, Hong Kong. The work of R. Khoshabeh was supported by the University of California, San Diego, under Program CalIT2 CSRO. The work of K. Gibson was support by the Space and Naval Warfare Systems Center Pacific (SSC Pacific) Naval Innovative Science and Engineering Program. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Sina Farsiu.

S. H. Chan, R. Khoshabeh, K. B. Gibson, and T. Q. Nguyen are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0112 USA (e-mail: h5chan@ucsd.edu).

P. E. Gill is with the Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2158229

Standard single-image restoration has been studied for more than half a 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 variational methods are also becoming mature [7]–[11].

While single-image restorations still have a room for improvement, we consider in this paper the video restoration problem. The key difference between an image and a video is the additional time dimension. Consequently, video restoration has some unique features that do not exist in an 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 a 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, a huge gain in speed can be realized.
3) *Temporal consistency*
   Temporal consistency is concerned about the smoothness of the restored video along the time axis. Although smoothing can be spatially performed (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 of a 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

There are many works on the problem of video restoration, particularly in the domain of video superresolution. In [19]–[21], video superresolution is formulated in a regularized least square minimization framework, in which the bilateral total variation (TV) is used as the regularization function. Later, in [22] and [23], the concept of kernel regression to the video restoration problem is applied. Similar approaches can be also found in [24], where Ng *et al.* considered isotropic TV as the a regularization function and modified (1) to incorporate the geometric warp caused by motion. In [25], Belekos *et al.* proposed a novel prior that utilizes the motion vector field in updating the regularization parameters so that the prior is both spatially and temporally adaptive to the data. A recent work by Chan and Nguyen [26] has considered a regularization function of the residue between the current solution and the motion compensated version of the previous solution.

It is worth noting that most of the aforementioned methods recover a video in a frame-by-frame basis.[1] Additionally, all of these methods assume that the blur kernel is spatially invariant. While this assumption is valid for many superresolution scenarios where multiple shots of the same object are used to fuse a higher resolution image, it is invalid when the blur is caused by object motions. As a result, they are unable to handle the spatially variant motion blur kernel.

Our proposed algorithm is inspired by the concept of "space–time volume," which is first introduced in the early 90s by Jähne [27], and rediscovered by Wexler, Shechtman, Caspi, and Irani [28], [29]. The idea of space–time volume is to stack the frames of a video to form a 3-D 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.

The main drawback of space–time minimization is that the size of a space–time volume is much larger than that of a single image (or five frames in the case of [25]). Therefore, the authors of [29] only considered a Tikhonov regularized least square minimization ([29, eq. (3)]) in which a closed-form solution exists. More sophisticated regularization functions such as TV and bilateral TV do not seem possible under this framework for these nondifferentiable functions are difficult to efficiently solve.

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

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

which is known as the TV/L2 minimization and

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

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

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

[1] A version of [25] is able to simultaneously process multiple frames, but in practice, it only supports five frames at once.

or the isotropic TV norm

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

where 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 ith component of the vector $\mathbf{f}$. More details on these two equations will be discussed in Section II-C.

The proposed algorithm is based on the augmented Lagrangian method, which is an old method that has recently drawn significant attention [10], [11], [30]. Most of the existing augmented Lagrangian methods for image restoration follow from Eckstein and Bertsekas' operator splitting method [31], which can be traced back to the work of Douglas and Rachford [32], and the proximal point algorithm by Rockafellar [33], [34]. Recently, the operator splitting method has been proven to be equivalent to the splitting Bregman iteration for some problems [35], [36]. 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.

1) We extend the existing augmented Lagrangian method to solve space–time TV minimization problems (2) and (3). Augmented Lagrangian method was previously used to image restoration only [10], [11].
2) Half-quadratic penalty parameter is updated according to constraint violation. This leads to faster rate of convergence, compared with methods using a fixed parameter [10].
3) Because of the space–time data structure, our proposed algorithm is able to handle spatially variant motion blur problems (object motion blur). Existing methods such as [19]–[26] are unable to do so.
4) Compared with [29], which is also a space–time minimization method, our method achieves TV/L1 and TV/L2 minimization quality, whereas [29] only achieves Tikhonov least square minimization quality.
5) In terms of speed, we achieve significantly faster computational speed, compared with 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 (PC) (MATLAB). This implies the possibility of real-time processing on a graphics-processing unit.
6) The proposed algorithm supports a wide range of applications.
   a) *Video deblurring*: With the assistance of frame rate up-conversion algorithms, the proposed method can remove spatially variant motion blur for *real* video sequences;
   b) *Video disparity*: Occlusion errors and temporal inconsistent estimates in the video disparity can be handled by the proposed algorithm without any modification;
   c) *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 [37], [38]. 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, and 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 3-D 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 3-D 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.,

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

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

B. Three-Dimensional Convolution

The 3-D convolution is a natural extension of the conventional 2-D convolution. Given space–time volume $f(x, y, t)$ and the blur kernel $h(x, y, t)$, the convolved signal $g(x, y, t)$ is given by $g(x, y, t) = f(x, y, t) * h(x, y, t) \stackrel{\text{def}}{=} \sum_{u,v,\tau} h(u, v, \tau) f(x-u, y-v, t-\tau)$. Convolution is a linear operation; therefore, 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 signal $f(x, y, t)$ to $g(x, y, t)$ following the rule, i.e.,

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

Assuming periodic boundaries [39], the convolution matrix $\mathbf{H}$ is a *triple* block-circulant matrix—it has a block-circulant structure, and within each block, there is a submatrix of block circulant with circulant block. Circulant matrices are diagonalizable using discrete Fourier transform (DFT) matrices [40], [41]:

Fact 1: If $\mathbf{H}$ is a triple block-circulant matrix, then it can be diagonalized by the 3-D DFT matrix $\mathbf{F}$ as

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

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

C. Forward-Difference Operators

We define operator $\mathbf{D}$ as a collection of three suboperators $\mathbf{D} = [\mathbf{D}_x^T \quad \mathbf{D}_y^T \quad \mathbf{D}_t^T]^T$, 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 suboperators are

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

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 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 $\mathbf{D} = [\beta_x\mathbf{D}_x^T \quad \beta_y\mathbf{D}_y^T \quad \beta_t\mathbf{D}_t^T]^T$.

With $(\beta_x, \beta_y, \beta_t)$, the anisotropic TV norm $\|\mathbf{f}\|_{\mathrm{TV1}}$ and the isotropic TV $\|\mathbf{f}\|_{\mathrm{TV2}}$ are defined according to (4) and (5), respectively. When $\beta_x = \beta_y = 1$ and $\beta_t = 0$, $\|\mathbf{f}\|_{\mathrm{TV2}}$ is the 2-D TV of $\mathbf{f}$ (in space). When $\beta_x = \beta_y = 0$ and $\beta_t = 1$, $\|\mathbf{f}\|_{\mathrm{TV2}}$ is the 1-D TV 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 $\|\mathbf{f}\|_{\mathrm{TV1}}$ is equivalent to the vector 1-norm on $\mathbf{Df}$, i.e., $\|\mathbf{f}\|_{\mathrm{TV1}} = \|\mathbf{Df}\|_1$. Therefore, for notation simplicity, we use $\|\mathbf{Df}\|_1$ instead. For $\|\mathbf{f}\|_{\mathrm{TV2}}$, although $\|\mathbf{f}\|_{\mathrm{TV2}} \neq \|\mathbf{Df}\|_2$ using the vector 2-norm definition, we still define $\|\mathbf{Df}\|_2 \stackrel{def}{=} \|\mathbf{f}\|_{\mathrm{TV2}}$ 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], [31]. Therefore, instead of repeating the details, we focus on the modifications made to the 3-D data structure. Additionally, our discussion is focused on the anisotropic TV, i.e., $\|\mathbf{Df}\|_1$. The isotropic TV, $\|\mathbf{Df}\|_2$ can be similarly derived.

A. TV/L2 Problem

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

$$\underset{\mathbf{f}}{\text{minimize}} \quad \frac{\mu}{2}\|\mathbf{Hf} - \mathbf{g}\|^2 + \|\mathbf{Df}\|_1 \quad (7)$$

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, i.e.,

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

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 (9), intermediate variable $\mathbf{u}$ and Lagrange multiplier $\mathbf{y}$ can be respectively partitioned as

$$\mathbf{u}=[\mathbf{u}_x^T \quad \mathbf{u}_y^T \quad \mathbf{u}_t^T]^T, \quad \text{and} \quad \mathbf{y}=[\mathbf{y}_x^T \quad \mathbf{y}_y^T \quad \mathbf{y}_t^T]^T. \tag{10}$$

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 to iteratively solve the following subproblems:

$$\begin{aligned}\mathbf{f}_{k+1} &= \arg\min_{\mathbf{f}} \frac{\mu}{2}\|\mathbf{Hf}-\mathbf{g}\|^2-\mathbf{y}_k^T(\mathbf{u}_k-\mathbf{Df}) \\ &\quad + \frac{\rho_r}{2}\|\mathbf{u}_k-\mathbf{Df}\|^2 &(11)\\ \mathbf{u}_{k+1} &= \arg\min_{\mathbf{u}} \|\mathbf{u}\|_1-\mathbf{y}_k^T(\mathbf{u}-\mathbf{Df}_{k+1}) \\ &\quad + \frac{\rho_r}{2}\|\mathbf{u}-\mathbf{Df}_{k+1}\|^2 &(12)\\ \mathbf{y}_{k+1} &= \mathbf{y}_k-\rho_r(\mathbf{u}_{k+1}-\mathbf{Df}_{k+1}). &(13)\end{aligned}$$

We now investigate these subproblems one by one.

1) $\mathbf{f}$-Subproblem: By dropping indexes k, the solution of problem (11) is found by considering the normal equation as follows:

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

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

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

where $\mathcal{F}$ denotes the 3-D Fourier transform operator. The matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, and $\mathcal{F}[\mathbf{H}]$ can be precalculated outside the main loop. Therefore, the complexity of solving (14) is in the order of $O(n\log n)$ operations, which is the complexity of the 3-D 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 a shrinkage formula [42]. Letting $\mathbf{v}_x=\beta_x\mathbf{D}_x\mathbf{f}+(1/\rho_r)\mathbf{y}_x$ (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 be also derived.

In case of isotropic TV, the solution is given by [42]

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

where $\mathbf{v}=\max\{\sqrt{|\mathbf{v}_x|^2+|\mathbf{v}_y|^2+|\mathbf{v}_t|^2},\epsilon\}$, and ϵ is a small constant ($\epsilon=10^{-6}$). Here, the multiplication and divisions are componentwise operations.

3) Algorithm: Algorithm 1 shows the pseudocode of the TV/L2 algorithm.

Algorithm 1 Algorithm for TV/L2 minimization problem

Input data $\mathbf{g}$ and $\mathbf{H}$.

Input parameters μ, β_x, β_y, and β_t.

Set parameters ρ_r (default $=2$) and α_0 (default $=0.7$).

Initialize $\mathbf{f}_0=\mathbf{g}$, $\mathbf{u}_0=\mathbf{Df}_0$, $\mathbf{y}=\mathbf{0}$, $k=0$.

Compute the matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, and $\mathcal{F}[\mathbf{H}]$.

while not converge **do**

1. Solve the $\mathbf{f}$-subproblem (11) using (15).
2. Solve the $\mathbf{u}$-subproblem (12) using (16).
3. Update the Lagrange multiplier $\mathbf{y}$ using (13).
4. Update ρ_r according to (24).
5. Check convergence:

if $\|\mathbf{f}_{k+1}-\mathbf{f}_k\|_2/\|\mathbf{f}_k\|_2\le \mathrm{tol}$ **then**

break

end if

end while

B. TV/L1 Problem

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

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

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

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 and $\sigma = 5$. Addition Gaussian noise is added to the image so that the BSNR is 40 dB.

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

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

which can be solved by considering the following normal equation:

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

yielding

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

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 (16).

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

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

Thus, using the shrinkage formula, the solution is

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

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

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

5) Algorithm: Algorithm 2 shows the pseudocode of the TV/L1 algorithm.

Algorithm 2 Algorithm for TV/L1 minimization problem

Input $\mathbf{g}$, $\mathbf{H}$, and parameters μ, β_x, β_y, and β_t. Let $k = 0$.
Set parameters ρ_r (default $= 2$), ρ_o (default $= 100$), and α_0 (default $= 0.7$).
Initialize $\mathbf{f}_0 = \mathbf{g}$, $\mathbf{u}_0 = \mathbf{Df}_0$, $\mathbf{y}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{Hf}_0 - \mathbf{g}$, and $\mathbf{z}_0 = \mathbf{0}$.
Compute matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$, $\mathcal{F}[\mathbf{D}_t]$, and $\mathcal{F}[\mathbf{H}]$.
while not converge **do**

1. Solve the $\mathbf{f}$-subproblem (19) using (20).
2. Solve the $\mathbf{u}$-subproblem (12) using (16).
3. Solve the $\mathbf{r}$-subproblem (21) using (22).
4. Update $\mathbf{y}$ and $\mathbf{z}$ using (23).
5. Update ρ_r and ρ_o according to (24).
6. Check convergence:

if $\|\mathbf{f}_{k+1} - \mathbf{f}_k\|_2/\|\mathbf{f}_k\|_2 \leq \text{tol}$ **then**

break

end if

end while

C. Parameters

In this subsection, we discuss the choice of parameters.

1) Choosing μ: The regularization parameter μ trades off the least square error and the TV 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 prior to solving the minimization. Recent advances in the operator-splitting methods have considered 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 to estimate the noise level for the noise characteristic of a video is never exactly known. Empirically, a reasonable μ for a natural image (and video sequence) typically lies in the range $[10^3, 10^5]$. Figs. 1 and 2 show the recovery results by using different values of μ. In the case of TV/L1 minimization, μ is typically lying in the range [0.1, 10].

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 and $\sigma = 1.10\%$ of the pixels are corrupted by salt and pepper noise. Image source: [43].

2) Choosing ρ_r: One of the major differences between the proposed algorithm and FTVd 4.0 [35][2] is the update of ρ_r. In [35], ρ_r is a fixed constant. However, as mentioned in [44], the method of multipliers can exhibit a faster rate of convergence by adapting the following parameter update scheme:

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

Here, condition $\|\mathbf{u}_{k+1} - \mathbf{D}\mathbf{f}_{k+1}\|_2 \geq \alpha\|\mathbf{u}_k - \mathbf{D}\mathbf{f}_k\|_2$ specifies the constraint violation with respect to constant α_r. The intuition is that the quadratic penalty $(\rho_r/2)\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$ is a convex surface added to the original objective function $\mu\|\mathbf{H}\mathbf{f} - \mathbf{g}\|^2 + \|\mathbf{u}\|_1$ so that the problem is guaranteed to be strongly convex [33]. Ideally, residue $(\rho_r/2)\|\mathbf{u}_k - \mathbf{D}\mathbf{f}_k\|^2$ should decrease as k increases. However, if $(\rho_r/2)\|\mathbf{u}_k - \mathbf{D}\mathbf{f}_k\|^2$ is not decreasing for some reasons, one can increase the weight of penalty $(\rho_r/2)\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$, relative to the objective, so that $(\rho_r/2)\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$ is forced to be reduced. Therefore, given α and γ, where $0 < \alpha < 1$ and $\gamma > 1$, (24) makes sure that the constraint violation is asymptotically decreasing. In the steady state, as $k \to \infty$, ρ_r becomes a constant [46]. The update for ρ_o in TV/L1 follows a similar approach.

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 $\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$ 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; otherwise, the effect of the quadratic surface $\|\mathbf{u} - \mathbf{D}\mathbf{f}\|^2$ 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 ratio (BSNR) is 40 dB. 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 (24), ρ_r is increased by a factor of γ if the condition is satisfied. Note that [35] (FTVd 4.0) is a special case when $\gamma = 1$, whereas the proposed algorithm allows the user to vary γ.

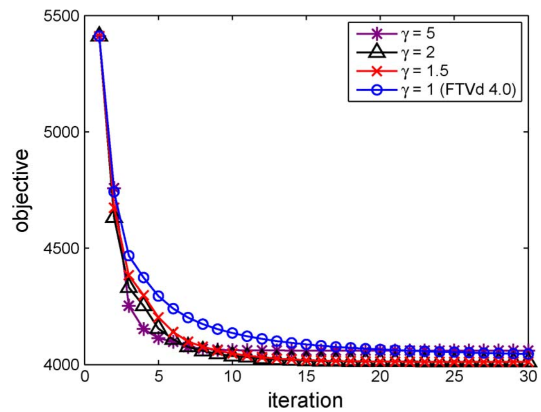


Fig. 3. Convergence profile of the proposed algorithm for deblurring the image "cameraman.tif". (Four colored curves) The rate of convergence using different values of γ, where γ is the multiplication factor for updating ρ_r.

In Fig. 3, the y-axis is the objective value $(\mu/2)\|\mathbf{H}\mathbf{f}_k - \mathbf{g}\|^2 + \|\mathbf{f}_k\|_{\mathrm{TV2}}$ for the kth iteration, and the x-axis is iteration number k. As shown in the figure, an appropriate choice of γ significantly improves the rate of convergence. 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 parameters ρ_r, γ, and α. In this test, 20 images are blurred by a Gaussian blur kernel of size 9×9, with variance $\sigma = 1$. The BSNR is 30 dB. For each image, two of the three parameters (ρ_r, γ, and α) are fixed at their default values, i.e., $\rho_r = 2$, $\gamma = 2$, and $\alpha = 0.7$, whereas one of them is varying within the range specified in Table I. The stopping criteria of the algorithm is $\|\mathbf{f}_{k+1} - \mathbf{f}_k\|_2/\|\mathbf{f}_k\| \leq 10^{-3}$, $\mu = 10^4$, and $(\beta_x, \beta_y, \beta_t) = (1, 1, 0)$ for all images. The maximum peak signal-to-noise ratio (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 20 images for ρ_r, γ, and α are 0.311, 0.208, and 0.357 dB, respectively. For an average PSNR difference in the order of 0.3 dB, the perceivable difference is small.[3]

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

[3]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.

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

Image no.	$1.5 \le \rho_r \le 10$			$1 \le \gamma \le 5$			$0.5 \le \alpha \le 0.9$		
	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

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.[4]

IV. APPLICATIONS

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

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

and apply a frame-by-frame approach to individually recover $\mathbf{f}_k$. In [26], 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 $\mathbf{f}_k$. Operators $\mathbf{D}_i$ are the spatial forward finite-difference operators oriented at angles 0°, 45°, 90°, and 135°. 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 [24], [25]

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

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 kth frame can be recovered by solving the following minimization [24]:

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

where a_i is a constant and $\|\mathbf{f}_k\|_{\mathrm{TV2}}$ is the isotropic TV on the kth frame. The method presented in [25] replaces the objective function by a weighted least squares and the isotropic TV regularization function by a weighted 2-norm on gradient. The weights are adaptively updated (using residue and motion vector field) in each iteration, and therefore, the regularization function is nonstationary, both spatially and temporally.

[4]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, $\sigma = 5$, and BSNR = 40 dB. The machine used is Intel Qual Core at 2.8 GHz, with 4-GB random access memory (RAM), and 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 and 1.28 s for FTVd 4.0 and the proposed method, respectively.

TABLE II
COMPARISONS BETWEEN THE OPERATOR-SPLITTING METHODS FOR TV/L2 MINIMIZATION

	Fast-TV (2008) [30]	FTVd 3.0 (2008) [10]	FTVd 4.0 (2009) [35] Split Bregman (2008) [45] 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[4]	83.39 sec	7.86 sec	2.94 sec	1.79 sec

TABLE III
COMPARISONS BETWEEN THE VIDEO RESTORATION METHODS

	Belekos 2010 [25]	Ng 2007 [24]	Chan 2011 [26]	Shechtman 2005 [29]	Proposed
Class of problem	super-resolution	super-resolution	deblurring	super-resolution	deblurring
Approach	multi-frames to multi-frames	frame-by-frame	frame-by-frame	space-time volume	space-time volume
Spatial Consistency	$\sum\limits_{i} \sum\limits_{d\in\{x,y\}} (\mathbf{D}_d\mathbf{f})^T\mathbf{A}_i^d(\mathbf{D}_d\mathbf{f})$	$\sum_i \sqrt{[\mathbf{D}_x\mathbf{f}]_i^2 + [\mathbf{D}_y\mathbf{f}]_i^2}$	$\sum_i \lVert\mathbf{D}_i\mathbf{f}\rVert_1$	$\lVert\mathbf{D}_x\mathbf{f}\rVert^2 + \lVert\mathbf{D}_y\mathbf{f}\rVert^2$	$\lVert\mathbf{f}\rVert_{TV2}$, Equation (5)
Temporal Consistency	$\sum_{i,j} \lVert\mathbf{f}_i - \mathbf{M}_{ij}\mathbf{f}_j\rVert^2_{\mathbf{B}_{ij}}$	$\lVert\mathbf{HM}_{ik}\mathbf{f}_k - \mathbf{g}_i\rVert^2$	$\lVert\mathbf{f}_k - \mathbf{M}_k\hat{\mathbf{f}}_{k-1}\rVert^2$	$\lVert\mathbf{D}_t\mathbf{f}\rVert^2$	$\lVert\mathbf{f}\rVert_{TV2}$, 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

A drawback of these methods is that the image recovery result heavily depends on the accuracy of motion estimation and compensation. In particular, in occlusion areas, the assumption that $\mathbf{M}_{i,k}$ is a one-to-one mapping [47] 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 $\lVert\mathbf{HM}_{i,k}\mathbf{f}_k - \mathbf{g}_i\rVert^2$ can lead to a 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 [24], 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, blur operator $\mathbf{H}$ is a block-circulant matrix. However, in the multichannel model, the operator $\mathbf{HM}_{i,k}$ is not a block-circulant matrix. The block-circulant property is a critical factor to speed as it allows the use of Fourier transform methods. For methods in [24] and [25], 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.

Our approach to solve spatially invariant blur problem shares the same insight as [29], which does *not* consider motion compensation. The temporal error is handled by spatio–temporal TV $\lVert\mathbf{Df}\rVert_2 = \sum_i \sqrt{|[\mathbf{D}_x\mathbf{f}]_i|^2 + |[\mathbf{D}_y\mathbf{f}]_i|^2 + |[\mathbf{D}_t\mathbf{f}]_i|^2}$. 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 TV, the temporal TV 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
PSNR, E_S, AND E_T VALUES FOR FOUR VIDEO SEQUENCES BLURRED BY GAUSSIAN BLUR KERNEL 9×9, $\sigma = 1$, AND BSNR = 30 dB

		"Foreman"	"Salesman"	"Mother"	"News"
PSNR (dB)	Blurred	28.6197	29.9176	32.5705	28.1106
	[29]	31.6675	33.0171	36.1493	34.0113
	[24]	32.5500	33.8408	38.2164	34.1207
	[26]	33.2154	33.8618	39.6991	34.7133
	Proposed	**33.7864**	**34.7368**	**40.0745**	**35.8813**
E_S $(\times 10^4)$	[29]	1.2067	1.1706	0.82665	1.3764
	[24]	1.1018	1.0743	0.71751	1.2146
	[26]	**1.0076**	**0.9934**	0.61544	1.123
	Proposed	1.0930	1.0105	**0.61412**	**1.1001**
E_T $(\times 10^3)$	[29]	10.954	3.3195	3.7494	4.6484
	[24]	10.827	2.4168	2.9397	3.7503
	[26]	10.202	2.5471	2.7793	3.3623
	Proposed	**9.3400**	**1.9948**	**2.0511**	**2.6165**

Table IV, and Figs. 4 and 5 show the comparisons between [24], [26], and [29] 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 BSNR is 30 dB.

The specific settings of the methods are as follows. For [29], we consider the following minimization:

$$\underset{\mathbf{f}}{\text{minimize}} \quad \mu\lVert\mathbf{Hf}-\mathbf{g}\rVert^2+\beta_x^2\lVert\mathbf{D}_x\mathbf{f}\rVert^2+\beta_y^2\lVert\mathbf{D}_y\mathbf{f}\rVert^2+\beta_t^2\lVert\mathbf{D}_t\mathbf{f}\rVert^2$$

and set the parameters empirically for the best recovery quality: $\mu = 200$ and $(\beta_x, \beta_y, \beta_t) = (1, 1, 1.25)$. For [24], instead of

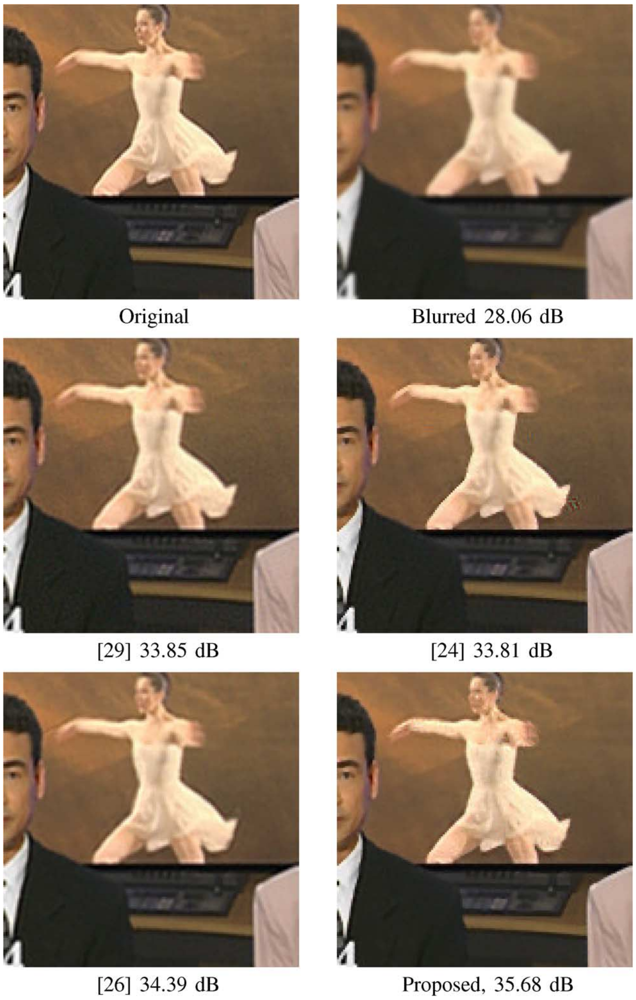


Fig. 4. "News" sequence; frame 100. (a) Original image (cropped for better visualization). (b) Blurred by a Gaussian blur kernel of size 9×9, $\sigma = 1$, and BSNR $= 30$ dB. (c)–(f) Results by various methods (see Table IV).

Fig. 5. "Salesman" sequence; frame 10. (a) Original image (cropped for better visualization). (b) Blurred by a Gaussian blur kernel of size 9×9, $\sigma = 1$, and BSNR $= 30$ dB. (c)–(f) Results by various methods (see Table IV).

using the CG presented in this 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 CG (LSQR [48]) to accommodate the nonblock-circulant operator $\mathbf{HM}_{i,k}$. The motion estimation is 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 [24] is set as 6 (out of 255), and the regularization parameter is $\lambda = 0.001$ [see (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 [26], the regularization parameters are also empirically chosen for the best recovery quality: $\lambda_S = 0.001$ and $\lambda_T = 0.05$.

To compare these methods, we apply TV/L2 (Algorithm 1) with the following parameters (same for all four videos): $\mu = 2000$ and $(\beta_x, \beta_y, \beta_t) = (1, 1, 1)$. All other parameters take the default setting: $\alpha = 0.7$, $\gamma = 2$, and $\rho_r = 2$. 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 performance of the algorithms. PSNR measures the image fidelity. Spatial TV 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 temporal TV E_T is defined as $E_T = \sum_i |[\mathbf{D}_t\mathbf{f}]_i|$ for each frame [26]. The average (over all frames) PSNR, E_S, and E_T are listed in Table IV.

Referring to the results, it is shown 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 [29] is equivalent to the 3-D Wiener deconvolution (regularized). Therefore, there exists a closed-form solution, but the result looks blurrier than the other methods. Among the four methods, both [24] and [26] use motion estimation and compensation. However, [24] is more sensitive to the motion estimation error—motion estimation error in some fast-moving areas are amplified in the deblurring step. Reference [26] is more robust to motion estimation error, but the computation time is significantly longer than the proposed method. The run time of [24] and [26] are approximately 100 s per frame (per color channel), whereas the proposed algorithm only requires approximately 2 s per frame (per color channel). These statistics are based on recovering videos of size 288×352, using a PC with Intel Qual Core at 2.8 GHz, with 4-GB RAM, and Windows 7/MATLAB 2010.

Fig. 6. "Market Place" sequence; frame 146. (Top) The original observed video sequences. (Middle) Result of [29]. (Bottom) Result of the proposed method.

Fig. 7. "Super Loop" sequence; frame 28. (Top) The original observed video sequences. (Middle) Result of [29]. (Bottom) Result of the proposed method.

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 [29] for detailed explanations). To this end, we apply [49] 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 empirically chosen as $\mu = 1000$ and $(\beta_x, \beta_y, \beta_t) = [1, 1, 5]$. 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 [29], using parameters $\mu = 1000$ and $(\beta_x, \beta_y, \beta_t) = [1, 1, 2.5]$.

As shown in Figs. 6 and 7, the proposed algorithm produces a significantly higher quality result than [29]. We also tested for a range of parameters μ and β for [29]. However, we observe that the results are either oversharpened (serious ringing artifacts) or undersharpened (not enough deblurring).

3) Limitation: The proposed algorithm requires considerably less memory than other TV minimization algorithms such as interior point methods. However, for high-definition videos, the proposed algorithm still has a 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.

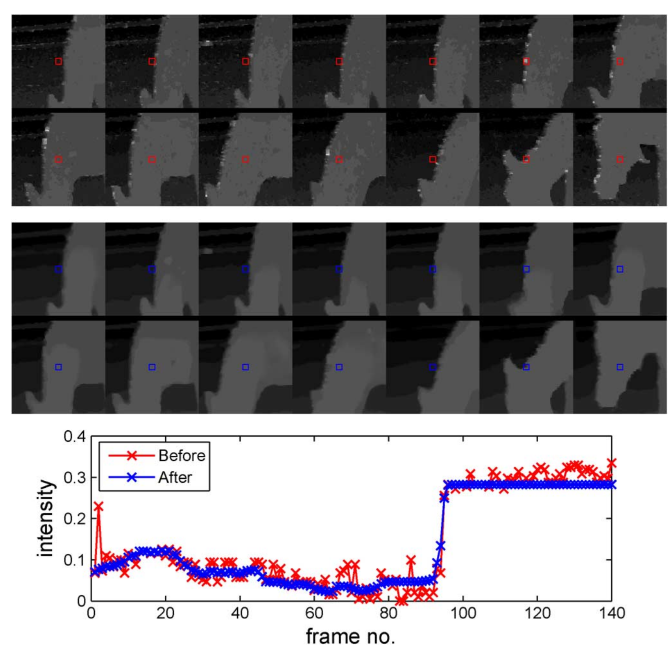
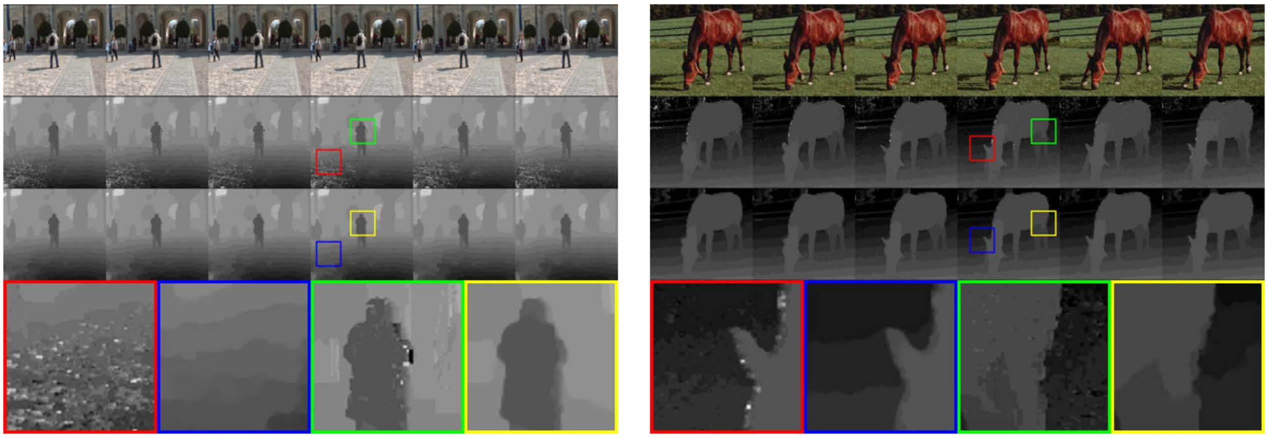


Fig. 8. (Top) Before applying the proposed TV/L1 algorithm. (Middle) After applying the proposed TV/L1 algorithm. (Bottom) Time evolution of the disparity value (normalized) of a pixel.

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 typically happens to areas with nonuniform and rapid motion.



(a) (b)

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 $\mu = 0.75$, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$, $\alpha = 0.7$, $\rho_r = 2$, $\rho_o = 100$, and $\gamma = 2$. (Last row) Zoom-in comparisons. (a) "Old Timers" sequence. (b) "Horse" sequence.

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 3-D space, saliency for stereo videos, stereo coding, and view synthesis

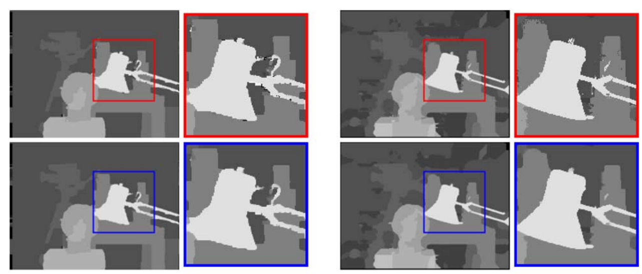
There are numerous papers on generating one disparity map based on a pair of stereo images [50]. However, all of these methods cannot be extended to videos because the energy functions are considered in a frame-by-frame basis. Although there are works in enforcing temporal consistency for adjacent frames, such as [51] and [52], 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 [53] and the cross-bilateralateral grid [54] 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, i.e.,

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

There are two reasons for choosing TV/L1 instead of TV/L2 in refining video disparity. First, disparity is a piecewise 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 1-norm $\|\mathbf{f} - \mathbf{g}\|_1$ is related to the notion of percentage of bad pixels, which is a quantity commonly used to evaluate disparity estimation algorithms [50]. Given a ground truth disparity $\mathbf{f}^*$, the number of bad pixels of an estimated disparity $\mathbf{f}$ is the cardinality of the set $\{i| \quad |[\mathbf{f} - \mathbf{f}^*]_i| > \tau\}$ 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 $\Omega_\tau = \{i| \quad |[\mathbf{f} - \mathbf{g}]_i| > \tau\}$, 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 nonconvex 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.



Algorithm no. 8 Algorithm no. 78

Fig. 10. Image disparity refinement on algorithms 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, $(\beta_x, \beta_y, \beta_t) = (1, 1, 0)$, $\alpha = 0.7$, $\rho_r = 2$, $\rho_o = 100$, and $\gamma = 2$.

2) Video Results: Two real videos ("Horse" and "Old Timers") are tested for the proposed algorithm. These stereo videos are downloaded from http://sp.cs.tut.fi/mobile3dtv/stereo-video/. Fig. 9 illustrates the results. The first row in Fig. 9 shows the left view of the stereo video. The second row shows the results of applying [53], [54] to the stereo video. Note that we are implementing a spatio–temporal version of [54], which uses adjacent frames to enhance the temporal consistency. However, the estimated disparity is still noisy, particularly 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), which is an area that [53] and [54] are unable to handle.

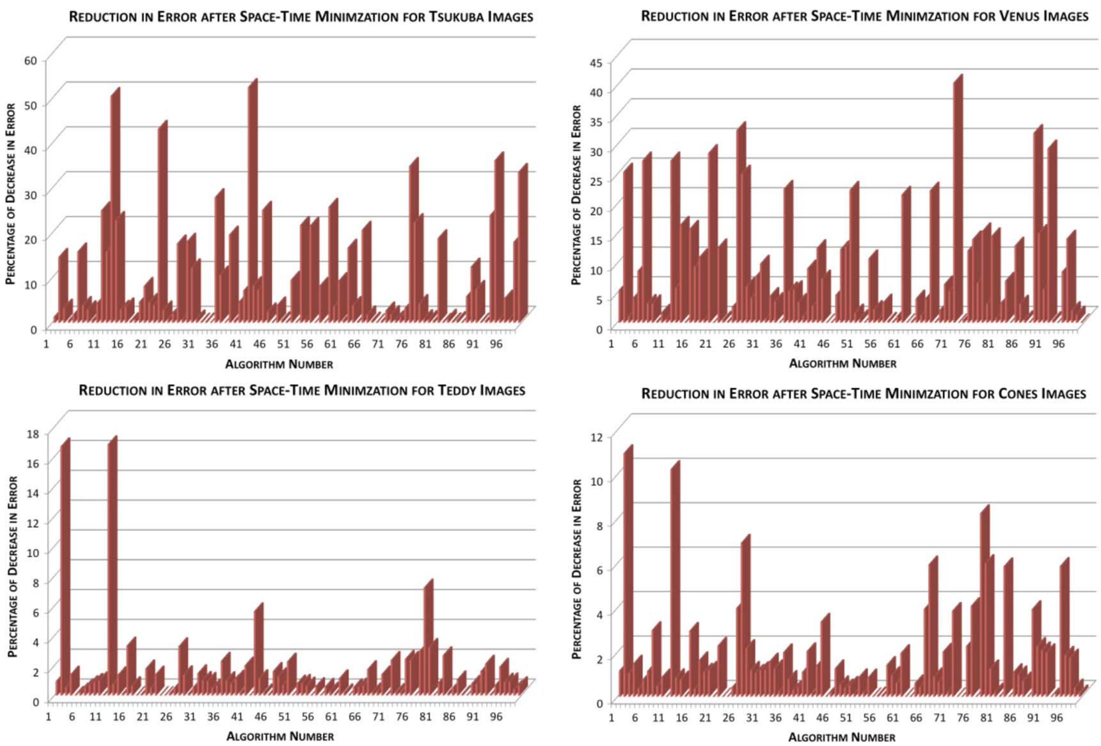


Fig. 11. Percentage error reduction (in terms of number of bad pixels) by applying the proposed algorithm to all the 99 methods on the Middlebury stereo database.

3) Image Results: The effectiveness of the proposed algorithm can be further elaborated by comparing to the 99 benchmark methods on Middlebury stereo evaluation website [50]. For all the 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 ready 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. 10 shows the results for two of the 99 methods (randomly chosen) for the data set "Tsukuba," and Fig. 11 shows the percentage of error reduction (in terms of the 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 is shown that the errors are typically reduced by a large margin of over 10%. While there is less error reduction for some data sets, it is important to note that error reduction is *always nonnegative*. In other words, the proposed algorithm always improves the initial disparity estimate. Furthermore, for *every* algorithm, we provide improvement in at least one of the image sets.

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 particularly happens 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 is 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 [55]. 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 [55] assumes that the refractive index is randomly changing but is also statistically stationary [56], [57]. 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 [60] and Shimizu *et al.* [61] proposed to first compensate the geometric distortion using nonrigid registration [62] and then deblur the images using deconvolution algorithms. The limitation is that nonrigid registration works well only when the geometric distortion can be adjusted by all the control points in the grid [62]. However, imaging through hot-air turbulence contains both large-area distortion (perceived as waving) and small disturbance (perceived as jittering). If nonrigid registration has to be used to compensate small disturbance, then the number of control points will be huge, making the computation not practical. There are other methods such as lucky frame/region fusion approach [63], [64]. However, these methods cannot effectively handle small disturbance 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

(a) input (b) step 1 [58], [59] (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 GLG [58], [59] to the input. (c) Step 2: Apply the proposed method to the results of Step 1.

Fig. 13. Zoom-in of "Acoustic Explorer" sequence; frames 25–28 (object is 2 mi from the camera). (Top) Input video sequence with contrast enhanced by GLG. (Bottom) Processed video by applying the proposed method to the output of GLG.

turbulence can be regarded as temporal noise, whereas the object movement is regarded as temporal edge. Under this framework, spatially invariant blur can be also incorporated. If the input video originally has a low contrast, a preprocessing step using gray level grouping (GLG) [58], [59] can be used (See Fig. 12).

2) Experiments: Fig. 13 shows the snapshots (zoom in) of a video sequence "Acoustic Explorer." In this example, GLG 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 empirically determined. Comparing the video quality before and after applying the proposed method, fewer jittering such as artifacts are observed in the processed videos. While this may not be apparent by viewing the still images, the improvement is significant in the 24 fps videos.[5]

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 clearly seen in the input sequence, but it becomes sharper in the processed sequence.

3) Limitation: The aforementioned experiments indicate that the proposed algorithm is effective for reducing small hot-air turbulence effects. However, for large-area geometric distortions, nonrigid 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.

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

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

V. Conclusion

In this paper, we have proposed a video deblurring/denoising algorithm that minimizes a TV 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 TVs 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*. Englewood Cliffs, NJ: Prentice-Hall, 2007.

[2] L. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, no. 6, pp. 745–754, Jun. 1974.

[3] W. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Amer.*, vol. 62, no. 1, pp. 55–59, Jan. 1972.

[4] V. Mesarovic, N. Galatsanos, and A. Katsaggelos, "Regularized constrained total least-squares image restoration," *IEEE Trans. Image Process.*, vol. 4, no. 8, pp. 1096–1108, Aug. 1995.

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

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

[7] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.

[8] T. Chan, G. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, no. 6, pp. 1964–1977, Nov. 1999.

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

[10] Y. Wang, J. Yang, W. Yin, and Y. Zhang, An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise CAAM, Rice Univ., Houston, TX, TR-0812, Sep. 2008.

[11] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[12] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[13] B. Lucas, "Generalized image matching by the method of differences," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1984.

[14] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph. (SIGGRAPH)*, vol. 27, no. 3, pp. 73:1–73:10, Aug. 2008.

[15] S. Dai and Y. Wu, "Motion from blur," in *Proc. IEEE CVPR*, 2008, pp. 1–8.

[16] A. Levin, "Blind motion deblurring using image statistics," in *Proc. NIPS*, 2006, pp. 841–848.

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

[18] J. Jia, "Single image motion deblurring using transparency," in *Proc. IEEE CVPR*, 2007, pp. 1–8.

[19] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[20] S. Farsiu, M. Elad, and P. Milanfar, "Multi-frame demosaicing and super-resolution of color images," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 141–159, Jan. 2006.

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

[22] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 550–563, Apr. 2008.

[23] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1958–1975, Sep. 2009.

[24] M. Ng, H. Shen, E. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," *EURASIP J. Adv. Signal Process.*, vol. 2007, pp. 1–16, 2007.

[25] S. Belekos, N. Galatsanos, and A. Katsaggelos, "Maximum a posteriori video super-resolution using a new multichannel image prior," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1451–1464, Jun. 2010.

[26] S. Chan and T. Nguyen, "LCD motion blur: Modeling, analysis and algorithm," *IEEE Trans. Image Process.* 2011 [Online]. Available: http://videoprocessing.ucsd.edu/~stanleychan/

[27] B. Jähne, *Spatio–Temporal Image Processing: Theory and Scientific Applications*. New York: Springer-Verlag, 1993.

[28] Y. Wexler, E. Shechtman, and M. Irani, "Space–time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[29] E. Shechtman, Y. Caspi, and M. Irani, "Space–time super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–545, Apr. 2005.

[30] Y. Huang, M. Ng, and Y. Wen, "A fast total variation minimization method for image restoration," *SIAM Multiscale Model Simul.*, vol. 7, no. 2, pp. 774–795, 2008.

[31] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[32] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc.*, vol. 82, no. 2, pp. 421–439, Jul. 1956.

[33] R. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Math. Oper. Res.*, vol. 1, no. 2, pp. 97–116, May 1976.

[34] R. Rockafellar, "Monotone operators and proximal point algorithm," *SIAM J. Control Optim.*, vol. 14, no. 5, pp. 877–898, 1976.

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

[36] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman UCLA, Los Angeles, CA, Tech. Rep. 09-31, 2009 [Online]. Available: ftp://ftp.math.ucla.edu/pub/camreport/cam09-31.pdf

[37] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented Lagrangian method for video restoration," in *Proc. IEEE ICASSP*, 2011.

[38] R. Khoshabeh, S. Chan, and T. Nguyen, "Spatio–temporal consistency in video disparity estimation," in *Proc. IEEE ICASSP*, 2011.

[39] B. Kim, "Numerical optimization methods for image restoration," Ph.D. dissertation, Dept. Manage. Sci. Eng., Stanford Univ., Stanford, CA, Dec. 2002.

[40] G. Golub and C. Van Loan, *Matrix Computation*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[41] M. Ng, *Iterative Methods for Toeplitz Systems*. London, U.K.: Oxford Univ. Press, 2004.

[42] C. Li, "An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing" M.S. thesis, Dept. Comput. Appl. Math., Rice Univ., Houston, TX, 2009.

[43] A. Levin, A. Rav-Acha, and D. Lischinski, "Spectral matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1699–1712, Oct. 2008.

[44] M. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed. New York: Academic, 1969, pp. 283–298.

[45] T. Goldstein and S. Osher, The split Bregman algorithm for L1 regularized problems UCLA, Los Angeles, CA, Tech. Rep. 08-29, 2008.

[46] D. Bertsekas, "Multiplier methods: A survey," *Automatica*, vol. 12, no. 2, pp. 133–145, Mar. 1976.

[47] M. Choi, N. Galatsanos, and A. Katsaggelos, "Multichannel regularized iterative restoration of motion compensated image sequences," *J. Vis. Commun. Image Represent.*, vol. 7, no. 3, pp. 244–258, Sep. 1996.

[48] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982.

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

[50] Middleburry Stereo Dataset [Online]. Available: http://bj.middlebury.edu/~schar/stereo/web/results.php

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

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

[53] K. Yoon and I. Kweon, "Locally adaptive support-weight approach for visual correspondence search," in *Proc. IEEE CVPR*, 2005, pp. 924–931.

[54] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proc. ECCV*, 2010, pp. 510–523.

[55] M. Roggemann and B. Welsh, *Imaging Through Turbulence*. Boca Raton, FL: CRC Press, 1996.

[56] J. Goodman, *Statistical Optics*. New York: Wiley-Interscience, 2000.

[57] J. Goodman, *Introduction to Fourier Optics*, 4th ed. Englewood, CO: Roberts & Company Publishers, 2004.

[58] 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. Image Process.*, vol. 15, no. 8, pp. 2290–2302, Aug. 2006.

[59] 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. Image Process.*, vol. 15, no. 8, pp. 2303–2314, Aug. 2006.

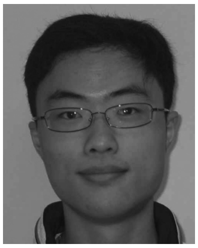
[60] X. Zhu and P. Milanfar, "Image reconstruction from videos distorted by atmospheric turbulence," *Vis. Inf. Process. Commun.*, vol. 7543, no. 1, p. 754 30S, 2010.

[61] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, "Super-resolution from image sequence under influence of hot-air optical turbulence," in *Proc. IEEE CVPR*, 2008, pp. 1–8.

[62] R. Szeliski and J. Coughlan, "Spline-based image registration," *Int. J. Comput. Vis.*, vol. 22, no. 3, pp. 199–218, Mar./Apr. 1997.

[63] D. Fried, "Probability of getting a lucky short-exposure image through turbulence," *J. Opt. Soc. Amer.*, vol. 68, no. 12, pp. 1651–1657, Dec. 1978.

[64] M. Aubailly, M. Vorontsov, G. Carhat, and M. Valley, "Automated video enhancement from a stream of atmospherically-distorted images: The lucky-region fusion approach," *Proc. SPIE*, vol. 7463, p. 746 30C, 2009.



Stanley H. Chan (S'06) received the B.Eng. degree in electrical engineering (with first class honors) from the University of Hong Kong, Pokfulam, Hong Kong, in 2007, and the M.A. degree in applied mathematics from the University of California, San Diego (UCSD), La Jolla, in 2009. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, UCSD.

His research interests include large-scale convex optimization algorithms, image and video restoration, spatially variant distortion, and space–time signal processing.

Mr. Chan is a Croucher Foundation Scholar from 2008 to 2010 and a UCSD Summer Graduate Teaching Fellow in 2011. He was also the recipient of the 2007 Sumida–Yawata Foundation Scholarship, the 2006–2007 Hong Kong Electric Company Ltd. Scholarship, the 2004 CLP Scholarship for Electrical Engineers, and the 2004 Electric Core and Manufacturing Ltd. Scholarship.

Ramsin Khoshabeh (S'08) received the B.S. and M.S. degrees in electrical engineering in 2005 and 2007, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla.

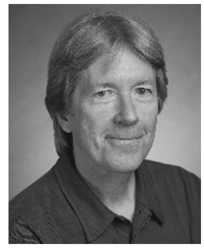
As an undergraduate, he specialized in computer design and graduated with highest honors (*summa cum laude*). In graduate school, his emphasis has been on computer vision and 3-D. His thesis work centers on the application of 3-D in the realm of surgical practice. His collaborators include two world-renowned surgeons and an expert in digital signal processing, all of which have guided him toward his academic goals. His interests include image and video processing, machine learning, robotics, biology and chemistry, and any opportunities in which he may help mankind through science and technology.

Mr. Khoshabeh was a recipient of California Institute of Telecommunications and Information Technology Strategic Research Opportunities Fellowship.

Kristofor B. Gibson (S'11) received the B.S. degree from Purdue University, West Lafayette, IN, in 2001, the M.S. degree, under the Science, Mathematics, and Research for Transformation Scholarship, in 2009 from the University of California, San Diego, La Jolla, where he is also currently working toward the Ph.D. degree.

He has been with Space and Naval Warfare Systems Center Pacific (SSC Pacific) as a Project Engineer since 2001 and, more recently, a Video and Image Surveillance Research Engineer. His research interests are image and video processing and enhancements in addition to computer vision algorithms and applications.

Mr. Gibson was the recipient of the AFCEA/U.S. Naval Institute 2010 Copernicus Award Winner for the successful development of a video-enhancement capability for the Navy.



Philip E. Gill received the Ph.D. degree in mathematics from the Imperial College of Science and Technology, University of London, London, U.K., in 1974.

He is currently a Distinguished Professor with the Department of Mathematics, University of California, San Diego, La Jolla. He has more than 30 years of experience in the development of optimization algorithms and the production of usable software. He is the coauthor of the computer packages NPSOL and SNOPT that have been distributed to thousands of universities, research laboratories, and industrial sites around the world. He lists three textbooks on optimization, the most recent of which is *Numerical Linear Algebra and Optimization* (Addison-Wesley, 1988). He has made more than 60 invited presentations at meetings around the world and lists over 100 publications in refereed journals or edited books. He works in the area of scientific computation, with special reference to numerical optimization.

Prof. Gill was the Cochair of the Sixth SIAM Conference on Optimization held in Atlanta, GA, in 1999. He has served in the Editorial Board for the *SIAM Journal on Optimization*, which is the *SIAM Journal on Matrix Analysis and Mathematical Programming Computation*, and in the organizing committee of eight national and international conferences on optimization.

Truong Q. Nguyen (F'05) received the B.S., M.S., and Ph.D. degrees in electrical engineering from California Institute of Technology, Pasadena, in 1985, 1986, and 1989, respectively.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla. He is the coauthor (with Prof. Gilbert Strang) of a popular textbook, i.e., "Wavelets and Filter Banks" (Wellesley-Cambridge Press, 1997), and the author of several MATLAB-based toolboxes on image compression, electrocardiogram compression, and filter bank design. His research interests are video processing algorithms and their efficient implementation.

Prof. Nguyen is currently the Series Editor (Digital Signal Processing) of Academic Press. He served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1994 to 1996, the Signal Processing Letters from 2001 to 2003, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1996 to 1997 and from 2001 to 2004, and the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2004 to 2005. He was the recipient of the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (Image and Multidimensional Processing area) for the paper that he has coauthored with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks in 1992. He was the recipient of the National Science Foundation Career Award in 1995.