LCD Motion Blur: Modeling, Analysis, and Algorithm

Stanley H. Chan, *Student Member, IEEE*, and Truong Q. Nguyen, *Fellow, IEEE*

***Abstract*—Liquid crystal display (LCD) devices are well known for their slow responses due to the physical limitations of liquid crystals. Therefore, fast moving objects in a scene are often perceived as blurred. This effect is known as the LCD motion blur. In order to reduce LCD motion blur, an accurate LCD model and an efficient deblurring algorithm are needed. However, existing LCD motion blur models are insufficient to reflect the limitation of human-eye-tracking system. Also, the spatiotemporal equivalence in LCD motion blur models has not been proven directly in the discrete 2-D spatial domain, although it is widely used. There are three main contributions of this paper: modeling, analysis, and algorithm. First, a comprehensive LCD motion blur model is presented, in which human-eye-tracking limits are taken into consideration. Second, a complete analysis of spatiotemporal equivalence is provided and verified using real video sequences. Third, an LCD motion blur reduction algorithm is proposed. The proposed algorithm solves an l_1-norm regularized least-squares minimization problem using a subgradient projection method. Numerical results show that the proposed algorithm gives higher peak SNR, lower temporal error, and lower spatial error than motion-compensated inverse filtering and Lucy–Richardson deconvolution algorithm, which are two state-of-the-art LCD deblurring algorithms.**

***Index Terms*—Human visual system, liquid crystal displays (LCDs), motion blur, subgradient projection, spatial consistency, temporal consistency.**

I. Introduction

LIQUID CRYSTAL display (LCD) devices are known to have slow responses due to the physical limitations of liquid crystals (LC). LC are organic fluids that exhibit both liquid and crystalline like properties. They do not emit light by themselves, but the polarization phase can be changed by electric fields [1]. A common circuit used in LCD to control the electric fields is known as the thin-film transistor (TFT) [2]. Although TFT responds quickly, it takes some time for the LC to change its phase. This latency is known as the fall time if the signal is changing from high to low, or the rise time if the signal is changing from low to high. Since the fall and rise times are not infinitesimal, the step response of an LC exhibits a sample-hold characteristic (see Fig. 1).

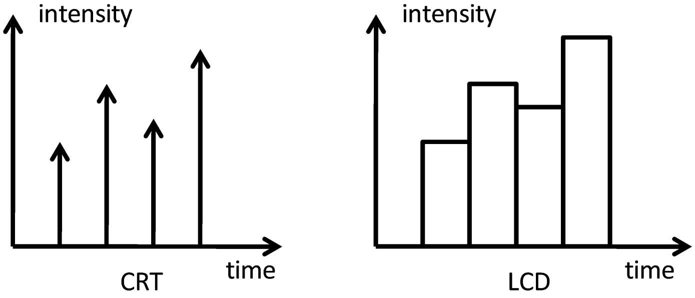


Fig. 1. Signaling characteristics of a cathode ray tube (CRT) and an LCD. CRT shows spontaneous response, whereas LCD demonstrates a sample-hold response.

Compared to LCD, traditional cathode ray tube (CRT) displays do not have the sample-hold characteristic. When a phosphor is exposed to electrons, it starts to emit light. As soon as the electrons leave, the phosphor stops emitting light. The latency of a phosphor is typically between 20 and 50 μs [2], but the time interval between two frames is 16.67 ms for a 60-frame per second video sequence. In other words, the latency of a phosphor becomes negligible compared to the frame interval.

Due to the sample-hold characteristic of LCs, fast moving scenes displayed on the LCD are often seen blurred. This phenomenon is known as the LCD *motion* blur. We emphasize the word "motion" because if the scene is stationary, LCD and CRT will give essentially the same degree of sharpness.

A. Review of Existing Methods

There are a number of methods to reduce LCD motion blur. Backlight flashing presented by Fisekovi *et al.* [3] is one of the earliest methods. In this method, the backlight (typically a cold cathode fluorescent lamp, CCFL) is controlled by a pulsewidth modulation [4]. Backlight flashing reduces motion blur but it also causes fluctuation in luminance. If the flashing rate is not high enough, the luminance fluctuation can be seen by human eyes, hence, causing eye strains. Therefore, in order to surpass the human eye limit (MPRT[1] 5.7 ms [6]), some advanced CCFL control methods are used, such as the active lamp technique presented by Yoon *et al.* [6].

Signal overdrive [7] is another commonly used method to reduce motion blur. The motivation to overdrive a signal is that the phase change of an LC is faster if the electric field is stronger. This phenomenon is explained in [1] and experimentally verified in [8]. Therefore, if the input signal is changing from 0 to 200 (in grayscale), then instead of sending a signal from 0 to 200, the overdrive circuit produces a signal from 0 to 210 (or a different value, depending on the circuit). Signal overdriving is

Manuscript received August 11, 2009; revised April 23, 2010, and August 23, 2010; accepted January 13, 2011. Date of publication January 31, 2011; date of current version July 15, 2011. This work was supported in part by the Croucher Foundation Scholarship and Samsung Information Systems America, Inc. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sabine Susstrunk.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093 USA (e-mail: h5chan@ucsd.edu; nguyent@ece.ucsd.edu).

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.2011.2109728

[1] MPRT stands for motion picture response time. [5]

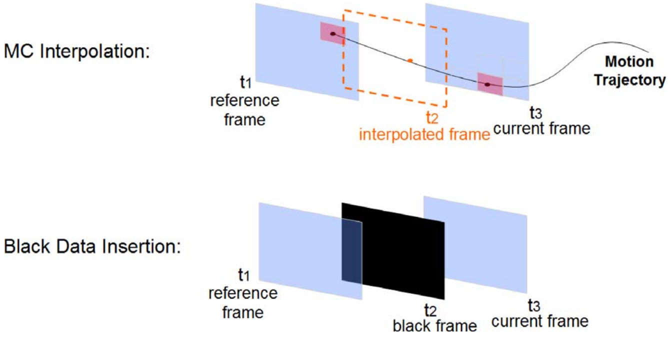


Fig. 2. Two commonly used frame rate up conversion (FRUC) method. Top: full frame insertion method by motion compensation (MC). Bottom: black frame insertion method.

often implemented using a lookup table, and a particular value is determined by the intensity change of a pixel. Image contents such as spatial and temporal consistencies are not considered.

Frame rate up conversion (FRUC) schemes is the third class of methods. The motivation of FRUC is that if the LC response can be improved, then the frame rate of LCD should also be increased. There are two major FRUC methods in the market: one is black frame insertion, as presented by Hong *et al.* [9], and the other one is full frame insertion presented in many papers such as [10]–[14]. Fig. 2 illustrates these two FRUC methods.

The last class of methods is the signal processing approach, in which the input signal is oversharpened so that it can compensate the motion blur caused by the LCD. Among all the methods, the motion-compensated inverse filtering (MCIF) techniques presented by Klompenhouwer and Velthoven [15] is the most popular one. MCIF first models motion blur as a finite impulse response (FIR) filter. Then, it finds an approximated inverse of the FIR filter to oversharpen the image. MCIF can also be used together with FRUC scheme, as presented in [16]. Another signal-processing approach is the deconvolution method proposed by Har-Noy and Nguyen [17]. In [17], the authors show that the deconvolution method gives better image quality than MCIF in terms of peak SNR (PSNR) and visual subjective tests.

B. Objectives and Related Work

There are three objectives of this paper: modeling, simulation, and algorithm.

First of all, we present a mathematical model for the hold-type LCD motion blur in the spatiotemporal domain. We do not consider the problem in the frequency domain as Klompenhouwer and Velthoven do in [15], because a video sequence is intrinsically a space-time signal [18]. It is more intuitive to study the motion blur in the spatiotemporal domain directly.

The modeling part of this paper is a generalization of [19]. In [19], Pan*et al.* show a fundamental equation for LCD motion blur modeling [(7) of [19]]. However, they implicitly assume that the human eyes are able to track objects perfectly. This is not true in general because our eyes have only limited range of tracking speed (See Section III). The same finding is reported by He *et al.* [20]. However, He *et al.* do not explain the cause of such a limit and they do not justify their MCIF design from a human visual system point of view. In contrast, our study of the eye-tracking limit is based on literature of cognitive science and verified using subjective tests.

The second objective of this paper is to provide a tool for the simulation of motion blur. A limitation of Pan's equation [(7) of [19]] is that the integration has to be performed in the temporal domain. To do so, the time step of the integration should be small, for otherwise, the integration cannot be approximated using a finite sum. Since the frame rate of a video sequence is fixed, in order to make the time step small, we need to interpolate intermediate frames. Temporal interpolation is time consuming: if the time step is 1/10 of the time interval between frames, then ten intermediate frames are needed. Therefore, the simulation of motion blur will be difficult unless there is an alternative method, which will be discussed in Section II.

The spatiotemporal equivalence has been used extensively in the literature but not proved. For example, Kurita [21] used the spatiotemporal equivalence to improve LCD image quality; Becker used the spatiotemporal equivalence to show the relation between blur edge width (BEW) and blur edge time for backlight scanning [4]; Tourancheau used the spatiotemporal equivalence to compare four commercially available LCD TVs [22]; Klompenhouwer showed the relation between BEW and frequency response of the blur operation [known as the temporal modulation transfer function (MTF)] [23]. Yet, none of these papers attempted to prove the spatiotemporal equivalence rigorously.

The most relevant paper in proving the spatiotemporal equivalence is [24]. Klompenhouwer drew a connection between the spatial and temporal apertures in a somewhat different—and very elegant—manner. However, a precise numerical approximation scheme for evaluating the continuous time integration in the discrete spatial domain is not pursued. Also, Klompenhouwer's paper is focused on the unit step input signal (which is a 1-D signal), whereas our study focuses on the general video signals.

The third objective of this paper is to propose a deconvolution algorithm based on the spatiotemporal equivalence.

A limitation of Klompenhouwer and Velthoven's MCIF [15] is that the MCIF cannot take into account of the spatial and temporal consistencies. Spatial consistency means that a pixel should have a value similar to its neighbors, unless it is along an edge in an image. Temporal consistency means that a pixel value should not change abruptly along the time axis, for otherwise, it will be seen as flickering artifacts. In this paper, we use a spatial regularization function to penalize variations in the spatial domain caused by noise. The l_1-normed regularization function used in our method is able to suppress the noise while preserving the edges. We also use a temporal regularization function to maintain the smoothness of the images along the time axis. In [25], Yao *et al.* proposed similar regularization functions in the context of coding artifacts removal. However, their problem setup is easier than ours because there is no blurring operators in their problem.

C. Organization

The organization of this paper is as follows. In Section II, we prove the spatiotemporal equivalence. We show by experiments that the spatial approximation to the temporal integration is accurate. In Section III, we present the findings of human-eye-tracking limits. Visual subjective tests are used to determine the optimal length of the FIR motion blur filter. In Section IV, we present the proposed algorithm. Comparisons with MCIF and Lucy–Richardson algorithm are discusses.

II. Spatiotemporal Equivalence

A. Review of LCD Motion Blur Model

For completeness, we provide a brief introduction to the LCD motion blur model. Most of the material presented in this section is due to Pan *et al.* [19].

Let $I_c(x,y,t)$ be a frame sampled at time t and suppose $I_c(x,y,t)$ has a motion vector (v_x, v_y). Let $h_D(t)$ be the step response of the display, where the subscript D can either be CRT or LCD. By Pan *et al.* [19], the image shown on the display is

$$I_s(x,y,t) = \int_{-\infty}^{\infty} h_D(\tau) I_c(x + v_x(t-\tau), y + v_y(t-\tau), t-\tau)\, d\tau. \quad (1)$$

An implicit assumption used in [19] is that the human-eye-tracking system is perfect, meaning that we can track any motion at any speed. Based on this, the motion compensated image formed on the retina becomes

$$\begin{aligned} I_m(x,y,t) &= I_s(x - v_x t, y - v_y t, t) \quad \text{(perfect eye tracking)} \\ &= \int_{-\infty}^{\infty} h_D(\tau) I_c(x - v_x\tau, y - v_y\tau, t-\tau)\, d\tau. \end{aligned} \quad (2)$$

Now assume that there is no low-pass filtering of the human visual system (HVS), then the observed signal becomes

$$I_o(x,y,y) = \int_{-\infty}^{\infty} h_D(\tau) I_c(x - v_x\tau, y - v_y\tau, t-\tau)\, d\tau. \quad (3)$$

To facilitate the discussion of this paper, we focus on the hold-type LCD. In this case, the step response of LCD is given by a boxcar signal, i.e., $h_{\mathrm{LCD}}(t) = 1/T$ for $0 \le t \le T$ and $h_{\mathrm{LCD}}(t) = 0$ for otherwise. With this setup, the image shown by an LCD is

$$I_o^{\mathrm{LCD}}(x,y,t) = \frac{1}{T}\int_0^T I_c(x - v_x\tau, y - v_y\tau, t-\tau)\, d\tau. \quad (4)$$

B. Proof of Spatiotemporal Equivalence

The integral in (4) can be evaluated by performing an integration over time $0 \le t \le T$. However, for a digitized version of the signal $I_c(x,y,t)$, there is no information between two consecutive frames. Therefore, it is never possible to compute the integral exactly. To alleviate this issue, an approximation scheme must be used. In the following, we discuss a spatiotemporal equivalence that allows us to approximate the temporal integration (4) by a spatial integration. But before we discuss the main theorem, we would like to provide some intuitive arguments.

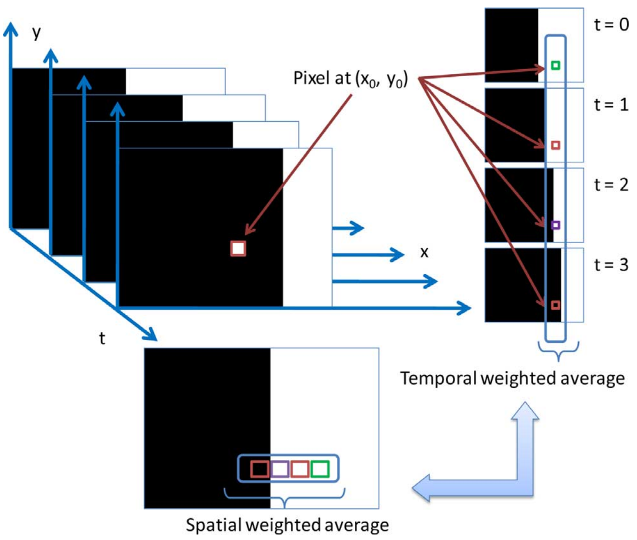


Fig. 3. Illustration of spatiotemporal equivalence. To evaluate the integral in (4), we first fix a position (x_0, y_0) and consider the pixel values at different times $t = 0, \ldots, 3$. The average is taken over the time, therefore, it is the average across the four marked pixels on the right-hand side. However, since these four frames are identical to each other (after motion compensation), we can evaluate the temporal average by averaging four adjacent pixels (in spatial domain).

Fig. 3 shows a video sequence. When integrating (4), we are essentially taking an average over the pixel values at a fixed position but at different time instants. Since all frames are highly correlated to each other (assume that there is no abrupt motions), we can approximate the average over different time instants as a spatial average over the pixel's neighborhoods. In this sense, we can transform the temporal average into a spatial average problem.

Definition 1: Given the velocities (v_x, v_y) and the sample-hold period T, we let $K \gg \max\{v_x T, v_y T\}$ be an integer multiple of $v_x T$ and $v_y T$, and define two sequences

$$\mathcal{S}_x = \left\{k, \quad \text{s.t. } \frac{k v_x T}{K} \text{ is an integer}, \text{ where } k \text{ is an integer}\right\}$$
$$\mathcal{S}_y = \left\{k, \quad \text{s.t. } \frac{k v_y T}{K} \text{ is an integer}, \text{ where } k \text{ is an integer}\right\}.$$

Define $\mathcal{S} = \mathrm{Sort}\{\mathcal{S}_x, \mathcal{S}_y\} = \{k,$ s.t. k is taken from $\mathcal{S}_x$ and $\mathcal{S}_y$ and k is sorted in an ascending order.

Define the weights $h(i,j)$ using the following algorithm:

For every $s_k \in \mathcal{S} = \{0, s_1, \ldots, s_P\}$,

1) If $s_k \in \mathcal{S}_x$, then $i \leftarrow i+1$, and $h(i,j) = (s_k - s_{k-1})/K$.
2) If $s_k \in \mathcal{S}_y$, then $j \leftarrow j+1$, and $h(i,j) = (s_k - s_{k-1})/K$.
3) $h(0,0) = s_1$.

Definition 1 is used to characterize the discrete running index and count the repeated indices, which will become clearer when we prove the theorem. As a quick example, consider $v_x T = 3$, $v_y T = 4$, and $K = 1200$. Using Definition 1, we have $\mathcal{S}_x = \{0, 400, 800, 1200\}$ and $\mathcal{S}_y = \{0, 300, 400, 600, 900, 1200\}$. If we concatenate these two sequences and sort them, then we have $\mathcal{S} = \{0, 300, 400, 600, 800, 900, 1200\}$. Thus, entries of $h(i,j)$ are

$$\begin{aligned} h(0,0) &= 300/1200 = 1/4 \\ h(0,1) &= (400-300)/1200 = 1/12 \\ h(1,0) &= (600-400)/1200 = 1/6 \\ &\vdots \quad \vdots \\ h(2,3) &= (1200-900)/1200 = 1/4. \end{aligned}$$

Theorem 1: Assume that $I_c(x,y,t) \approx I_c(x,y,t+\delta t)$ for $\delta t < T$. Let T be the sample-hold period of the LC, and $K \gg \max\{v_xT, v_yT\}$ be an integer multiple of v_xT and v_yT. Also, let M and N be the largest integer smaller than $v_xT(K-1)/(K)$ and $v_yT(K-1)/(K)$, respectively, i.e.,

$$M = \left\lfloor v_xT\frac{K-1}{K} \right\rfloor \qquad N = \left\lfloor v_yT\frac{K-1}{K} \right\rfloor$$

where $\lfloor \cdot \rfloor$ is the floor operator. Then, the integral (4) can be evaluated as follows:

$$\begin{aligned} I_o^{\mathrm{LCD}}(x,y,t) &= \frac{1}{T}\int_0^T I_c(x - v_x\tau, y - v_y\tau, t-\tau)\, d\tau \\ &\approx \sum_{i=0}^{M}\sum_{j=0}^{N} I_c(x-i, y-j, t) h(i,j) \qquad (5) \end{aligned}$$

where $h(i,j)$ is defined in Definition 1.

Proof: We first explain the assumption that $I_c(x,y,t) \approx I_c(x,y,t+\delta t)$ if $\delta t < T$. Digital video is a sequence of temporally sampled images of a continuous scene. Unless the scene contains extremely high-frequency components, such as a checkerboard pattern, typically the correlation between frames is high. Since no intermediate image is captured between two consecutive frames, we assume that $I_c(x,y,t) \approx I_c(x,y,t+\delta t)$ if $\delta t < T$. Other assumptions about the intermediate images are also possible, such as a linear translation from frame $I_c(x,y,t)$ to $I_c(x,y,t+T)$. But for simplicity, we assume that $I_c(x,y,t)$ holds until the next sample arrives.

Using this assumption, we have

$$\begin{aligned} I_o^{\mathrm{LCD}}(x,y,t) &= \frac{1}{T}\int_0^T I_c(x - v_x\tau, y - v_y\tau, t-\tau)\, d\tau \\ &\approx \frac{1}{T}\int_0^T I_c(x - v_x\tau, y - v_y\tau, t)\, d\tau. \qquad (6) \end{aligned}$$

Let $K \gg \max\{v_xT, v_yT\}$ be an integer multiple of v_xT and v_yT. Also, we let the finite difference interval be $\Delta\tau = (T/K)$. Then, the integral in (6) can be approximated by a finite sum

$$\begin{aligned} I_o^{\mathrm{LCD}}(x,y,t) &\approx \frac{1}{T}\int_0^T I_c(x - v_x\tau, y - v_y\tau, t)\, d\tau \\ &\approx \frac{1}{T}\sum_{k=0}^{K-1} I_c(x - v_xk\Delta\tau, y - v_yk\Delta\tau, t)\,\Delta\tau \\ &= \frac{1}{K}\sum_{k=0}^{K-1} I_c\left(x - k\frac{v_xT}{K}, y - k\frac{v_yT}{K}, t\right). \qquad (7) \end{aligned}$$

Now assume that $I_c(x,y,t)$ is a digital image at a particular time t. Since the image is composed of a finite number of pixels and each pixel has a finite size, we have $I_c(x,y,t) = I_c(x+\Delta x, y+\Delta y, t)$ if $|\Delta x| < 1$ and $|\Delta y| < 1$. Therefore, the above sum can be partitioned into groups as follows:

$$\begin{aligned} &I_o^{\mathrm{LCD}}(x,y,t) \\ &= \frac{1}{K}\sum_{k=0}^{K-1} I_c\left(x - k\frac{v_xT}{K}, y - k\frac{v_yT}{K}, t\right) \\ &= \frac{1}{K}\sum_{k=0}^{K-1} I_c\left(x - \left\lfloor k\frac{v_xT}{K}\right\rfloor, y - \left\lfloor k\frac{v_yT}{K}\right\rfloor, t\right) \\ &= \frac{1}{K}\left[\sum_{k=0}^{s_1-1} I_c(x,y,t) + \sum_{k=s_1}^{s_2-1} I_c(x - i_{1k}, y - j_{1k}, t) + \cdots \right. \\ &\quad \left. + \sum_{k=s_{P-1}}^{s_P} I_c(x - i_{P-1,k}, y - j_{P-1,k}, t)\right]. \end{aligned}$$

where $\mathcal{S} = \{s_1, \ldots, s_P\}$ is defined by Definition 1. In each term $\sum_{k=s_p}^{s_{p+1}-1} I_c(x - i_{pk}, y - j_{pk}, t)$, the indices i_{pk} (similarly for j_{pk}) are given by

$$i_{pk} = \begin{cases} \left\lfloor k\frac{v_xT}{K}\right\rfloor, & \text{if } s_p \in \mathcal{S}_x, \\ 0, & \text{otherwise.} \end{cases}$$

Using the definition of $h(i,j)$ in Definition 1, we can further simplify the above expression as follows:

$$\begin{aligned} &I_o^{\mathrm{LCD}}(x,y,t) \\ &\approx \frac{1}{K}\left[\sum_{k=0}^{s_1-1} I_c(x,y,t) + \sum_{k=s_1}^{s_2-1} I_c(x - i_{1k}, y - j_{1k}, t) + \cdots \right. \\ &\quad \left. + \sum_{k=s_{P-1}}^{s_P} I_c(x - i_{P-1,k}, y - j_{P-1,k}, t)\right] \\ &= \sum_{i=0}^{M}\sum_{j=0}^{N} I_c(x-i, y-j, t) h(i,j) \end{aligned}$$

where $M = \lfloor v_xT\frac{K-1}{K}\rfloor$ and $N = \lfloor v_yT\frac{K-1}{K}\rfloor$. ■

As explained earlier, the importance of Theorem 1 is that the *temporal* problem is transformed into a *spatial* problem. Therefore, the temporal motion blur can now be treated as spatial blur problem.

C. Example

To illustrate the meaning of the parameters in Theorem 1, we show an example. Suppose that there is a diagonal motion of $v_x = 180$ pixel per second and $v_y = 180$ pixel per second, and let us assume that the LCD has a sample-hold period of $T = 1/60$ s. Since $\max\{v_xT, v_yT\} = 3$, we may define $K = 6$ (K and is an integer multiple of $\max\{v_xT, v_yT\}$). Let $k = 0, 1, 2, 3, 4, 5$, then $i = 0, 1, 2$ and $j = 0, 1, 2$ (See Definition 1), and $M = 2$ and $N = 2$.

We define $S_x = \{0, 2, 4, 6\}$ and $S_y = \{0, 2, 4, 6\}$. Concatenating and sorting $\mathcal{S}_x$ and $\mathcal{S}_y$ yields $\mathcal{S} = \{0, 2, 4, 6\}$. Therefore,

1) $h(0,0) = (2-0)/6 = 1/3$;
2) $h(1,1) = (4-2)/6 = 1/3$;

3) $h(2,2) = (6-4)/6 = 1/3$;
4) $h(i,j) = 0$ for otherwise.

Thus, the observed LCD signal can then be computed as follows:

$$\begin{aligned} I_o^{\mathrm{LCD}}(x,y,t) &\\ \approx \sum_{i=0}^{M}\sum_{j=0}^{N} I_c(x-i,y-j,t)h(i,j) &\\ = I_c(x,y,t)h(0,0) + I_c(x-1,y-1,t)h(1,1) &\\ + \cdots + I_c(x-2,y-2,t)h(2,2) &\\ = \frac{1}{3}\left[I_c(x,y,t) + I_c(x-1,y-1,t)\right. &\\ \left. + I_c(x-2,y-2,t)\right]. & \end{aligned}$$

D. Discussion

There are some observations regarding Theorem 1.

First, Theorem 1 shows that although the perceived LCD blur is a temporal average, it can be approximated by a spatial average.

Second, the skewness of $h(i,j)$ is determined by the direction of the motion. If $v_x = v_y$ (as in our example), then $h(i,j)$ becomes diagonal; if $v_x = 0$, then $h(i,j)$ becomes vertical; and if $v_y = 0$, then $h(i,j)$ becomes horizontal. In these three special cases, all the nonzero entries of $h(i,j)$ are identical. If the motion direction is not horizontal, vertical, or diagonal, then an entry of $h(i,j)$ is larger if the distance between the line along the motion direction and (i,j) is closer.

Third, magnitude of the motion determines the length of the filter $h(i,j)$, hence, the blurriness of the perceived image. If there is no motion, then $h(i,j) = 1$, and therefore, there will be no blur. However, if the motion is large, then $h(i,j)$ will be long, and therefore, the averaging effect will be strong.

Fourth, compared to a 60-Hz LCD monitor, a 240-Hz LCD monitor shows better perceptual quality because it refreshes four times faster than a 60-Hz monitor. This effect can be reflected by reducing the sample-hold period T and hence the length of the filter $h(i,j)$.

E. Numerical Implementation of Theorem 1

Algorithm 1 Compute $h(i,j)$ and $I_o^{\mathrm{LCD}}(x,y,t)$

```
Fix a time instant $t$, and LCD decay time $T$.
Step 1: Use motion estimation algorithm to detect $(v_x, v_y)$.
Step 2: Define weights $h(i,j)$ according to Definition 1.
Step 3: Set $h(i,j) = 0$, if $i > L$ or $j > L$ for some $L$ (to
be discussed in Section III).
Step 4: Compute $I_o^{\mathrm{LCD}}(x,y,t)$ using via discrete
convolution in (5).
```

Algorithm 1 is a pseudocode for numerical implementation of Theorem 1. The algorithm consists of four steps. In the first step, motion vectors are computed using methods such, as full search, three-step search [26], directional methods [27], or hybrid methods [28]. The second step is to define the blur kernel $h(i,j)$ according to definition (1). Note that each $h(i,j)$ is defined locally, meaning that one motion vector defines one $h(i,j)$. If there is a collection of motion vectors, then correspondingly there will be a collection of $h(i,j)$. In step 3, $h(i,j)$ is limited to a finite length and width for modeling the eye-tracking property, which will be discussed in Section III. Lastly, the output can be computed via a discrete convolution shown in (5).

F. Comparison Between Spatial and Temporal Integration

To verify Theorem 1, we compare the integration (4) and spatial integration (5) using simulations. Our simulation methodology follows from [29], where the authors show that the simulation is a good substitute for a comprehensive experiment to measure LCs response.

Fig. 4 shows four simulation results. [2] For each video sequence, two consecutive frames are collected, and the relative motion is computed using a full search algorithm [26]. Ten motion-compensated frames are inserted via standard H.264 motion-compensation algorithm. This is to simulate a continuous time signal. The temporal integration is calculated as the average of the ten motion-compensated frames.

To measure the difference between spatial and temporal integration, PSNR values are computed (see Table I). As shown, on an average, the PSNR is higher than 40 dB, which implies a small difference between the two methods. However, the computing time using the spatial approximation is significantly shorter than the temporal integration (we used a 10× FRUC by linear interpolation).

III. Eye Movement Limit

In Section II, we assume that our eye-tracking system is perfect, i.e., we can track moving objects at any speed. This assumption makes the derivation simpler, but it is not true in reality. A more realistic model is that our eyes have a speed limit. We provide supports to this argument through the literature in cognitive science and visual subjective tests.

A. Eye Tracking

In Rayner's review [30] on eye-tracking system, he mentions that when we look at a scene, our eyes are rapidly moving. The rapid movement is known as the saccades, which can be as high as 500° s^{-1}. However, at such a high speed, we can hardly see any visual content. This phenomenon is known as the saccade suppression [31], [32]. Therefore, most of the images perceived are obtained during a period of time (typically about 200–300 ms) between saccades. This period is known as the fixation. If an object is moving quickly, then the duration of fixation is shortened, and hence, the perceptual quality reduces. Therefore, even if our eyes may be able to *track* an object, we may not be able to *see* what it is.

The relation between object speed and perceived sharpness can be concluded from the following findings.

1) Westerink and Teunissen [33] conducted two experiments about the relation between perceptual sharpness and the picture speed. In their first experiment, they asked the

[2] Complete set of videos are available online at http://videoprocessing.ucsd.edu/~stanleychan

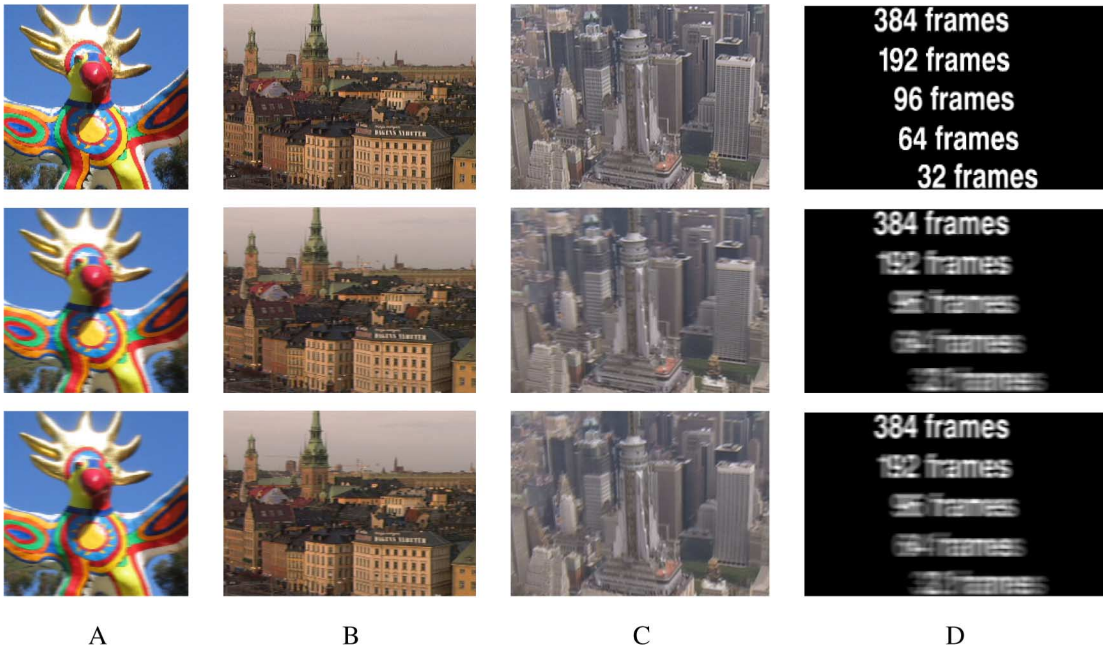


Fig. 4. Simulation results of spatial and temporal integration. Top row: original input image; middle row: simulated blur using spatial integration; and bottom row: simulated blur using temporal integration.

TABLE I
COMPARISON BETWEEN SPATIAL INTEGRATION AND TEMPORAL INTEGRATION. MAXIMUM MV REFERS TO THE MAXIMUM MOTION VECTOR IN THE IMAGE. PSNR MEASURES THE DIFFERENCE BETWEEN THE SPATIAL INTEGRATION TO THE TEMPORAL INTEGRATION. HIGHER PSNR IMPLIES SMALLER DIFFERENCE

Image	Size	Maximum MV	PSNR
A	200 × 200	4.35	42.45dB
B	640 × 480	3.71	41.34dB
C	320 × 240	7.23	41.10dB
D	300 × 600	10	40.81dB

viewers to track a moving image with their heads stay at a fixed position (referred to as the fixation condition). The conclusion is that the perceived sharpness drops to a minimum score when picture speed is beyond 5° s^{-1} ([33, Fig. 4]). A similar conclusion can also be drawn from [34].

2) In the second experiment by Westerink and Teunissen [33], viewers were allowed to move their heads (referred to as the pursuit condition). The conclusion is that the perceived sharpness drops to a minimum score when picture speed is beyond 35° s^{-1} ([33, Fig. 6]).
3) Bonse [35] studied a mathematical model for temporal subsampling. They mentioned that there is a maximum eye-tracking velocity of 5°–50° s^{-1}, which had been experimentally justified by Miller and Ludvigh [36].
4) Glenn and Glenn [37] studied the discrimination of human eyes on televised moving images of high resolution (300 line) and low resolution (150 line). Their results show that it is harder for human eye to discriminate high- from low-resolution images if the speed increases.
5) Gegenfurtner *et al.* [38] studied the relation between pursuit eye movement and perceptual performance. The viewers were asked to track a moving image of speed 4° s^{-1}. Results show that the recorded the eye velocities are ranged between 3° and 4.5° s^{-1}.

The conclusion of these findings is that when picture motion increases, the perceptual sharpness decreases. In some experiments, the maximum picture speed is found to be 5° s^{-1} for fixation condition, and 35° s^{-1} for pursuit condition. Beyond this threshold, our eyes are unable to capture visual content from the image.

B. LCD Model With Eye Tracking

The existence of the maximum eye-tracking speed implies that the LCD model has to be written as follows:

$$\begin{aligned} I_m(x,y,t) &= I_s(x-u_xt, y-u_yt, t) \\ &= \int_0^T h_D(\tau) I_c(x - v_x\tau - (u_x - v_x)t, \ldots \\ &\quad y - v_y\tau - (u_x - v_x)t, t - \tau)\, d\tau \end{aligned}$$

where u_x and u_y are the eye-tracking speed. If the picture speed is low, then our eyes are able to capture the visual content, and hence, $u_x = v_x$ and $u_y = v_y$. However, if the picture speed is beyond the threshold, then the difference $(u_x - v_x)\tau$ accounts for the images that we cannot see.

Consequently, we apply this observation to design inverse filters to reduce LCD motion blur. Previous efforts in inverse filter design for LCD motion blur can be found in [15], [17], and [39]. In these papers, the inverse filter is designed according to the estimated point-spread function $h(i,j)$. If $h(i,j)$ has a narrow frequency support, then noise in an image will be amplified by the inverse filter.

Due to the presence of the maximum eye-tracking speed, we know that fast moving objects cannot be seen clearly. Therefore, a natural question is that whether it is necessary to construct a very long $h(i,j)$ and let its inverse filter to introduce flickering

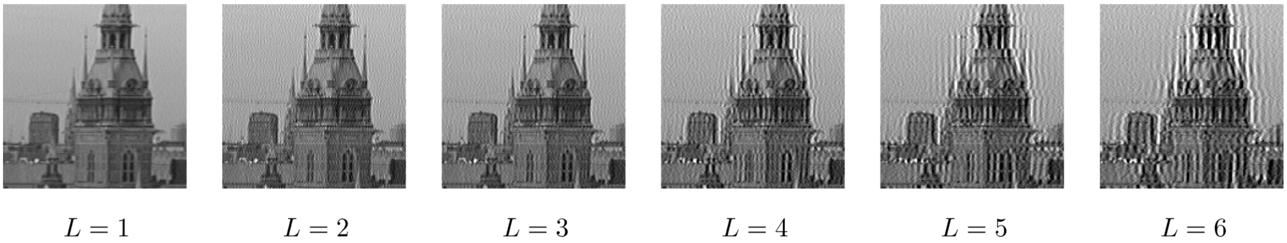

$L=1$ $L=2$ $L=3$ $L=4$ $L=5$ $L=6$

Fig. 5. Video 2 Stockholm. The sequence is processed using [39], with different values of L.

TABLE II
AVERAGE TV ERROR [DEFINED IN (8)] AROUND ADJACENT PIXELS

L	Video 1 Shield	Video 2 Stockholm	Video 3 Black White
1	0.0465	0.0546	0.0478
2	0.0589	0.0764	0.0504
3	0.0655	0.0895	0.0533
4	0.0896	0.1205	0.0608
5	0.1010	0.1279	0.0581
6	0.1146	0.1412	0.0620

TABLE III
SUBJECTIVE TESTS TO DETERMINE THE MAXIMUM LENGTH L

Subjective Test to determine optimal L			
	Video 1 Shield	Video 2 Stockholm	Video 3 Black White
mean	4.19	3.89	3.39
std	0.67	0.65	0.90

artifacts. To this end, we find that it is more appropriate to limit the size of $h(i,j)$ as follows:

$$h(i,j)=\begin{cases} h(i,j), & \text{if } i \le L \text{ and } j \le L \\ 0, & \text{else}\end{cases}$$

where L denotes the maximum number of pixels along the horizontal and vertical directions. For example, $L=4$ means that the size of $h(i,j)$ is at most 4 × 4 pixels.

The exact value of L is difficult to determine as it depends on a number of factors such as the conditions of 5° s^{-1} for fixation and 35° s^{-1} for pursuit. To compromise this issue, we seek a method to estimate a value of L so that it can be used for our deblurring algorithm, which will be described next.

C. Experiments

To determine the maximum length of the filter $h(i,j)$, we performed a visual subjective test.

Three video sequences are used in this test, where each video sequence consists of a global horizontal motion. The motion vectors are determined by full search algorithm, and the point-spread function $h(i,j)$ is found using Algorithm 1. In order to determine the maximum length L for $h(i,j)$, we truncate $h(i,j)$ using six different values of L. For each L, we oversharpen the video sequence by using the optimization approach presented in [39]. The optimization problem is solved using a conjugate gradient algorithm (LSQR [40]), with damping constant $\lambda = 1e^{-1}$. Maximum number of iterations is set to be 100, and tolerance level is set to be $1e^{-6}$.

Fig. 5 shows the results. When L increases, it can be observed that more artifacts are introduced. To quantify the amount of artifacts, we calculate the average total variation (TV) around neighborhood pixels

$$e=\left(\frac{1}{MN}\sum_{i,j}|f(i+1,j)-f(i,j)|^2 + |f(i,j+1)-f(i,j)|^2\right)^{1/2} \tag{8}$$

where $f(i,j)$ is the image under consideration, and M and N are the number of columns and rows of $f(i,j)$, respectively. Table II shows the total variation error.

The visual subjective test procedure follows from International Telecommunication Union Radiocommunication Sector (ITU-R) BT. 1082, Section 8 [41]. Eighteen human viewers were invited to the experiment. For each of the three video sequences, there are six levels of the maximum lengths ($L = 1,\ldots,6$). $L=1$ means that $h(i,j)$ is a delta function, which in turn implies that there is no inverse filtering. $L=6$ means that $h(i,j)$ has a size of 6 × 6, and therefore, there is a substantial inverse filtering. Each time, the viewers were presented a reference and a processed video sequence simultaneously. They were asked to tell whether the processed one showed any distracting artifacts. If they replied no, then L would be increased until the level such that noise became appealing. The videos were played on a PC with 2.8-GHz CPU, 8-GB DDR2 RAM, ATI Radeon 2600 XT 512 MB video card. The video sequences were uncompressed, played at 60 frames per second.

The mean and variance of L is shown in Table III. It can be observed that if we limit the size of the point-spread function $h(i,j)$ to 4 × 4 (on average) and apply the conjugate gradient algorithm to deblur the image, viewers can perceive the maximum degree of sharpness before they notice artifacts.

A limitation of this experiment is that it relies on the formulation in [39]. If other formulations such as the spatial and temporal regularization functions (See Section IV) are used, the maximum length L can possibly be increased as artifacts can be suppressed more using these methods.

IV. Deblurring Algorithm

The objective of this section is to propose a deblurring algorithm for LCD motion blur reduction.

A. Optimization Formulation

First, by spatiotemporal equivalence (5), we know that the observed (blurred) image is related to the original (sharp) image by a linear convolution. Therefore, we can apply the standard imaging model (see, e.g., [42]) to model the image formation as follows:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \eta \tag{9}$$

where $\mathbf{f} = \mathrm{vec}[f(x,y)]$ and $\mathbf{g} = \mathrm{vec}[g(x,y)]$ are vectors that denote the sharp image $f(x,y)$ and the observed (blurred) images $g(x,y)$ respectively. Here, $\mathrm{vec}(\,\cdot\,)$ is the vectorization operator, which stacks an image into a long column vector, according to the lexicographical order. $\mathbf{H}$ is a block circulant matrix denoting the blurring (convolution) operator, and η is an additive noise term.

The LCD deblurring problem may be formulated within an optimization framework by considering the least-squares minimization problem

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \|\mathbf{H}\mathbf{f} - \mathbf{g}\|_2^2 \\ \text{subject to} \quad & 0 \leq \mathbf{f} \leq 1. \end{aligned} \tag{10}$$

where $\|\cdot\|_2$ denotes the l_2-norm. The choice of l_2-norm is based on the assumption that the noise η is Gaussian. The bounds on the optimization variable $\mathbf{f}$ is to ensure that a pixel value does not exceed the range of $[0, \quad 255]$ or $[0, \quad 1]$ in the normalized scale.

Problem (10) is ill-posed because the operator $\mathbf{H}$ often has a large condition number. Therefore, in the presence of noise, solving (10) may lead to undesirable images. To resolve this issue, the standard method is to introduce a regularization function $\mathbf{R}_{\mathrm{reg}}(\mathbf{f})$ and solve

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \|\mathbf{H}\mathbf{f} - \mathbf{g}\|_2^2 + \lambda \mathbf{R}_{\mathrm{reg}}(\mathbf{f}) \\ \text{subject to} \quad & 0 \leq \mathbf{f} \leq 1. \end{aligned} \tag{11}$$

In statistics, the regularization is also known as the prior information about the image. The constant λ is a regularization parameter that weights the objective function relative to the regularization term.

B. Spatial Regularization

The spatial regularization function is defined by the gradients of the image. Specifically, we define the directional gradient operators $\mathbf{D}_x$, $\mathbf{D}_y$, $\mathbf{D}_{d1}$, and $\mathbf{D}_{d2}$ as follows:

$$\begin{aligned} \mathbf{D}_x\mathbf{f} &= \mathrm{vec}[f(x+1,y) - f(x,y)] \\ \mathbf{D}_y\mathbf{f} &= \mathrm{vec}[f(x,y+1) - f(x,y)] \\ \mathbf{D}_{d1}\mathbf{f} &= \mathrm{vec}[f(x-1,y+1) - f(x,y)] \\ \mathbf{D}_{d2}\mathbf{f} &= \mathrm{vec}[f(x+1,y+1) - f(x,y)] \end{aligned}$$

where $\mathbf{f} = \mathrm{vec}[f(x,y)]$ is the unknown image, $\mathbf{D}_x$ and $\mathbf{D}_y$ represent the directional derivative operators along the horizontal and vertical directions, respectively, and $\mathbf{D}_{d_1}$ and $\mathbf{D}_{d_2}$ represent the directional derivative operators along the direction from top left to bottom right and from top right to bottom left, respectively. The transposes of these operators are as follows:

$$\begin{aligned} \mathbf{D}_x^T\mathbf{f} &= \mathrm{vec}[f(x-1,y) - f(x,y)] \\ \mathbf{D}_y^T\mathbf{f} &= \mathrm{vec}[f(x,y-1) - f(x,y)] \\ \mathbf{D}_{d1}^T\mathbf{f} &= \mathrm{vec}[f(x+1,y-1) - f(x,y)] \\ \mathbf{D}_{d2}^T\mathbf{f} &= \mathrm{vec}[f(x-1,y-1) - f(x,y)]. \end{aligned}$$

The spatial regularization function is defined as follows:

$$\mathbf{R}_S(\mathbf{f}) = \sum_i \|\mathbf{D}_i\mathbf{f}\|_1 \tag{12}$$

where the subscript $i \in \{x, y, d_1, d_2\}$ represents the direction. This spatial regularization is a special case of the bilateral TV introduced by Farsiu *et al.* [43]–[45]. It can also be considered as an l_1 approximation to the conventional TV regularization introduced by Rudin *et al.* [46]. In [25], Yao *et al.* used a regularization function similar to ours for the application of removing coding artifacts.

The advantage of using the proposed spatial regularization over the conventional Tikhonov regularization $\mathbf{R}(\mathbf{f}) = \|\mathbf{D}_x\mathbf{f}\|^2 + \|\mathbf{D}_y\mathbf{f}\|^2$ is that Tikhonov regularization cannot preserve sharp edges. Fig. 6 shows some comparisons between the proposed spatial regularization and Tikhonov regularization. Detailed discussions can be found in [47] and [48].

C. Temporal Regularization

Although the spatial regularization function can be applied to each frame of a video individually, the temporal consistency of the video is not guaranteed. Temporal consistency describes whether two adjacent frames have a smooth transition. If a pixel has a sudden increase/decrease in brightness along the time axis, then it is said to have temporal inconsistency. As an illustration, two consecutive frames taken from a real video are shown in Fig. 7. Note that pixels around the edges of the window have different intensities in the two adjacent frames, although they are at the same location.

To enhance the temporal consistency, we introduce a regularization function along the temporal direction. A similar approach was previously used by Yao *et al.* for denoising [25]. The temporal regularization function is defined as follows:

$$\mathbf{R}_T(\mathbf{f}) = \|\mathbf{f} - \mathbf{M}\mathbf{f}_0\|_2^2$$

where $\mathbf{M}$ is a geometric wrap (i.e., motion compensation), and $\mathbf{f}_0$ is the solution of the previous frame. The interpretation of $\mathbf{R}_T(\mathbf{f})$ is that the current solution should be close to the previous solution after motion compensation. Thus, by minimizing $\mathbf{R}_T(\mathbf{f})$, we can reduce the temporal noise.

The effectiveness of the proposed temporal regularization function can be seen in Fig. 7. Fig. 7(a) and (b) shows two

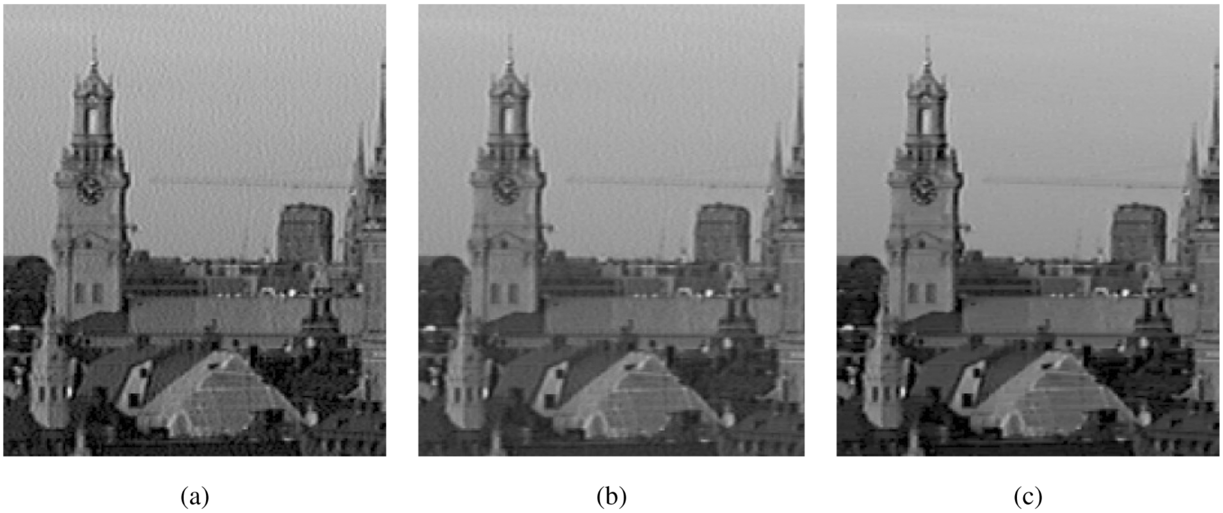


Fig. 6. Comparison between various regularization functions. (a) Solution obtained by minimizing $\|\mathbf{Hf}-\mathbf{g}\|_2^2$. (b) Solution obtained by Tikhonov $\|\mathbf{Hf}-\mathbf{g}\|_2^2 + \lambda\|\mathbf{D_{xf}}\|_2^2$, where $\lambda = 0.0005$. (c) Solution obtained by minimizing the proposed method $\|\mathbf{Hf}-\mathbf{g}\|_2^2 + \lambda\|\mathbf{D_{xf}}\|_1$, where $\lambda = 0.0015$.

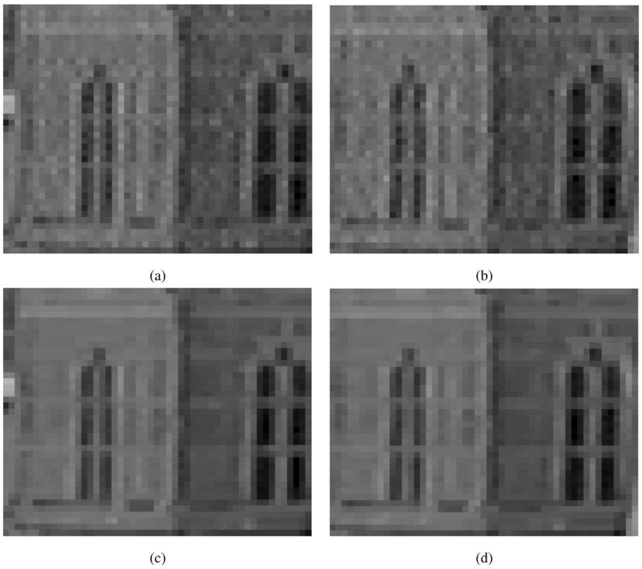


Fig. 7. Two consecutive frames. (a), (b) No temporal regularization. (c), (d) With temporal regularization.

consecutive frames without temporal regularization, where as Fig. 7(c) and (d) shows two consecutive frames with temporal regularization. It can be observed that the transition of pixel values is smoother in (c) and (d) than (a) and (b).

D. Convolution Operator $\mathbf{H}$

The convolution operator $\mathbf{H}$ is constructed based on the motion vectors. If the motion is global, then $\mathbf{H}$ corresponds to a spatially invariant point-spread function. In this case, $\mathbf{H}$ is a block-circulant-with-circulant-block (BCCB) matrix [49], and it can be diagonalized by Fourier transforms [50]. As a result, computation of the matrix–vector product $\mathbf{Hf}$ can be performed in $O(n \log n)$ operations, where n is the number of pixels.

For general video sequences, the motion is not global, and therefore, $\mathbf{H}$ does not correspond to a spatially invariant point-spread function. In the worst case, where every pixel has a different motion, each pixel will have a different point-spread function. Because of this, $\mathbf{H}$ does not have the BCCB structure, and therefore, it cannot be diagonalized by Fourier transforms. Hence, to compute the matrix–vector multiplication $\mathbf{Hf}$, one has to do it in the spatial domain directly. The complexity is in the order of $O(nk)$, where n is the number of image pixels, and k is the number of pixels of the largest point-spread function.

Since the motion is not global in general, many existing algorithms cannot be used as they assume $\mathbf{H}$ to a BCCB matrix. These methods include the half quadratic penalty methods by Huang *et al.* [51], Wang *et al.* [52], Geman and coauthors [53], [54], and Yao *et al.* [25], the interior point method by Nesterov [55], and the projected gradients methods by Chambolle [56]. In the following, we present a method that supports both BCCB matrices and general matrices.

E. Subgradient Projection Algorithm

The overall optimization problem is

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \|\mathbf{Hf}-\mathbf{g}\|_2^2 + \lambda \sum_i \|\mathbf{D}_i\mathbf{f}\|_1 + \gamma\|\mathbf{f}-\mathbf{Mf}_0\|_2^2 \\ \text{subject to} \quad & 0 \le \mathbf{f} \le 1 \end{aligned} \tag{13}$$

where λ and γ are two regularization parameters.

Subgradient projection is a variation of the steepest descent algorithm. Given the kth iterate, the algorithm updates the $k+1$th iterate by

$$\mathbf{f}^{k+1} = \mathbf{f}^k - \alpha_k \nabla \Bigg(\|\mathbf{Hf}^k - \mathbf{g}\|_2^2 + \cdots \\ + \lambda \sum_i \|\mathbf{D}_i \mathbf{f}^k\|_1 + \gamma\|\mathbf{f}^k - \mathbf{Mf}_0\|_2^2 \Bigg)$$

where α_k is the step size, and $\nabla(\cdot)$ is the gradient operator. Since the l_1 term is not differentiable, we consider its subgradient instead of the gradient. The (sub)gradients of individual terms are

$$\nabla\left(\|\mathbf{Hf}-\mathbf{g}\|_2^2\right)=2\mathbf{H}^T(\mathbf{Hf}-\mathbf{g}) \tag{14}$$

$$\nabla\left(\sum_i\|\mathbf{D}_i\mathbf{f}\|_1\right)=\sum_i\mathbf{D}_i^T\mathrm{sgn}(\mathbf{D}_i\mathbf{f}) \tag{15}$$

$$\nabla\left(\|\mathbf{f}-\mathbf{Mf}_0\|_2^2\right)=2(\mathbf{f}-\mathbf{Mf}_0) \tag{16}$$

where $\mathrm{sgn}(x)=1$ if $x>0$, -1 if $x<0$, and 0 if $x=0$.

The simple bound constraints can be handled by projecting out-of-bound components to their closest bounds. In other words, we set

$$[\mathbf{f}]_i=\begin{array}{ll}1, & \text{if } [\mathbf{f}]_i\geq 1\\ 0, & \text{if } [\mathbf{f}]_i\leq 0.\end{array} \tag{17}$$

where $[\cdot]_i$ denotes the ith component of $\mathbf{f}$.

The step size is chosen to satisfy the "square summable but not summable" rule (see, e.g., [57]–[59])

$$\sum_{k=0}^{\infty}\alpha_k^2<\infty \qquad \sum_{k=0}^{\infty}\alpha_k=\infty.$$

In our problem, we choose $\alpha_k=M/(M+k)$, for some maximum number of iterations M, typically $M=1000$.

We also implemented the Armijo line search algorithm [60], [61]. For fixed constants $0<\eta<1$, and $0<\rho<1$, we let $\Phi(\mathbf{f})=\|\mathbf{Hf}-\mathbf{g}\|_2^2+\lambda\sum_i\|\mathbf{D}_i\mathbf{f}\|_1+\gamma\|\mathbf{f}-\mathbf{Mf}_0\|_2^2$. If $\Phi(\mathbf{f}^{k+1})-\Phi(\mathbf{f}^k)\leq\alpha\eta\|\nabla\Phi(\mathbf{f}^k))\|_2^2$, then the step size α is reduced by $\alpha\leftarrow\rho\alpha$, until the condition is satisfied.

In theory, subgradient projection algorithm with the square summable rule has provable convergence [57], [59]. But in practice, if we allow the algorithm to terminate early, then the Armijo line search algorithm often gives better PSNR than square summable rule.

Algorithm 2 shows the pseudocode for our projected subgradient algorithm using the Armijo line search.

Algorithm 2 Subgradient Projection Algorithm

```
Set $\lambda$ and $\gamma$ (Typically, $\lambda \approx 0.0015$, $\gamma \approx 0.1$).
Set initial step size $\alpha_k = 1$.
Initialize variables.
while $k \leq k_{\max}$ do
    Compute the gradients $\nabla\Phi(\mathbf{f})$ as defined in (14)–(16).
    Armijo Line Search to determine step size $\alpha$.
    Update $\mathbf{f}^{k+1} = \mathbf{f}^k - \alpha\nabla\Phi(\mathbf{f}^k)$

        $\mathbf{f}^{k+1} = \begin{array}{ll} 1 & \text{if } \mathbf{f}^{k+1} > 1 \\ 0 & \text{if } \mathbf{f}^{k+1} < 0 \end{array}$

    $k \leftarrow k + 1$
end while
```

Regarding the regularization parameters λ and γ, Bertsekas [58] mentioned that these parameters can never be known prior to solving the problem. There are some methods to estimate the parameters, such as generalized cross validations by Nguyen *et al.* [62], or the L-curve criteria discussed in Hansen's book [63]. But these methods are not guaranteed to work for the nondifferentiable l_1 term. Therefore, in this paper, we test the images with a sequence of λ and γ, and choose the ones that balance PSNR, run time, and perceptual quality. In fully automated settings, an updating strategy based on a nonreference metric [64] can be used.

F. Experiments

In this section, we compare the performance of the proposed spatiotemporal deblurring algorithm versus existing algorithms. In particular, we measure three quantities of the deblurred signal.

1) Mean Square Error: The first quantity is the PSNR, which is defined as follows:

$$\mathrm{PSNR}=10\log_{10}\frac{1}{\mathrm{MSE}}$$

where MSE is the mean square error, defined as follows

$$\mathrm{MSE}=\frac{1}{MN}\|\mathbf{Hf}-\mathbf{g}\|_2^2$$

where M,N are the number of rows and columns of the image, respectively, and $\mathbf{f}$ is the minimization solution. PSNR measures the solution fidelity, and higher PSNR implies that the difference between $\mathbf{Hf}$ and $\mathbf{g}$ is smaller.

2) Spatial Consistency: Spatial consistency is a qualitative measurement of the deviation between neighborhood pixels. To quantify the spatial consistency, we define

$$E_S=\sum_i\|\mathbf{D}_i\mathbf{f}\|_1.$$

This quantity measures the TV of the solution $\mathbf{f}$. If E_S is large, then it is likely that $\mathbf{f}$ is noisy.

3) Temporal Consistency: Temporal consistency describes the smoothness of the video along the time axis. Given two consecutive frames $\mathbf{f}_{t+1}$ and $\mathbf{f}_t$, and the motion vector field, we define

$$E_T=\|\mathbf{f}_{t+1}-\mathbf{M}_t\mathbf{f}_t\|_2^2$$

where $\mathbf{M}_t$ is a geometric warping operator such that $\mathbf{M}_t\mathbf{f}_t$ is the motion-compensated frame with respect to $\mathbf{f}_{t+1}$.

4) Results: We ran two experiments, both are panning camera scenes. The videos have global horizontal motion blur, with some small local motions.

The specification of the video is as follows: the size is 640×480 and it is stored as a sequence of 8-bit grayscaled bit maps, therefore, each pixel has a dynamic range of 256 levels. For better numerical stability, we normalize the image by dividing the pixel values by 255. The video is supposed to be played at 60 fps, with 300 frames in total. We ran our experiment on a PC with AMD Dual Core 3 GHz, 8-GB RAM, Radeon-HD2600XT graphics card, Windows XP-64 OS.

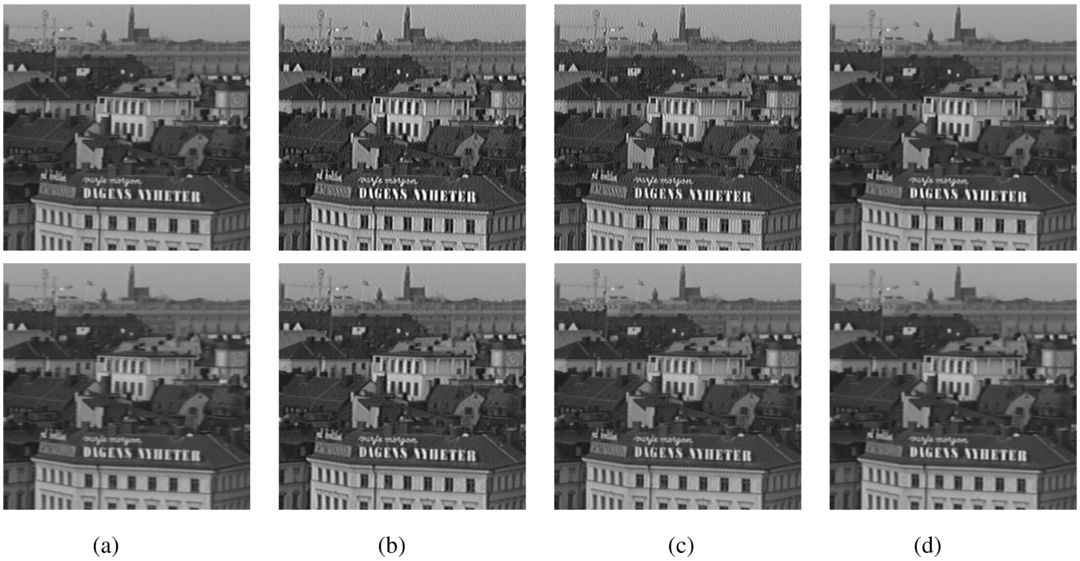


Fig. 8. Experiment 1: the upper row shows the synthesized signal that is sent to the LCD. The lower row shows the (simulated) perceived LCD signal. (a) Original signal. (b) Signal synthesized by MCIF [15]. (c) Signal synthesized by Lucy–Richardson [17]. (d) Signal synthesized by proposed method.

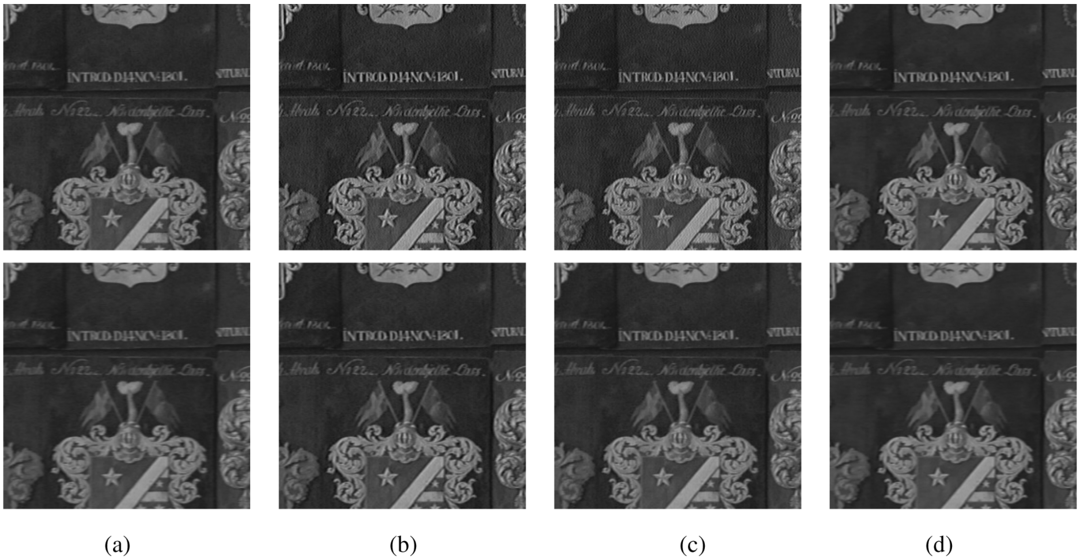


Fig. 9. Experiment 2: the upper row shows the synthesized signal that is sent to the LCD. The lower row shows the (simulated) perceived LCD signal. (a) Original signal. (b) Signal synthesized by MCIF [15]. (c) Signal synthesized by Lucy–Richardson [17]. (d) Signal synthesized by proposed method.

The results are shown in Figs. 8 and 9. The upper rows of the figures show the signals synthesized by different methods, namely MCIF [15], LR [17], and the proposed method. As shown, the synthesized signals of MCIF and LR contain a lot of noise. These noise are often inconsistent in time, and so when the images are moving, viewers will see flickering artifacts. In contrast, the proposed method controls the amount of noise, both spatially and temporally. Flickering is suppressed significantly.

The lower row of the figures show the simulated images that an viewer would see. We emphasize that these are simulated images because the actual images formed on the retina of a viewer are never accessible. To simulate the observed signal, we apply $\mathbf{H}$ to the synthesized signal $\mathbf{f}$.

Numerical results using PSNR, E_S and E_T are given in Table IV. Although the proposed method does not have a PSNR as high as Lucy Richardson, it shows a 2 dB improvement to the original input images. More important observations are the spatial consistency and the temporal consistency: the proposed method yields significantly lower error than the other two methods.

It should be noted that although our regularization functions has a better performance than existing methods in preserving edges, suppressing noise and enhancing temporal consistency, restoration of texture areas is still challenging. In areas where the magnitude of texture gradient is comparable to the magnitude of noise gradient, our current algorithm has limited performance in removing the noise while keeping the texture. Our future research is to develop methods to restore texture areas.

G. Visual Subjective Test

We ran a visual subjective test to verify our results. The subjective test is based on the single stimulus non-categorical judgment method described in ITU-R BT.500-11 [65]. In this

TABLE IV
COMPARISONS BETWEEN MCIF, LUCY–RICHARDSON, AND THE PROPOSED METHOD

		Signal to Noise Ratio	Spatial Error	Temporal Error
Video Name	Methods	PSNR (dB)	$\sum_i \|\mathbf{D}_i\mathbf{f}\|_1$	$\|\mathbf{f}_{t+1} - \mathbf{M}_t\mathbf{f}_t\|_2^2$
Stockholm	Original	34.43	4.8286×10^3	3.2969×10^1
	MC Inverse Filter [15]	34.357	9.8488×10^3	4.3412×10^2
	Lucy Richardson [17]	40.35	1.0914×10^4	5.5423×10^3
	Proposed Method	36.38	4.1443×10^3	3.3400×10^1
Shield	Original	36.879	3.586×10^3	7.0169×10^2
	MC Inverse Filter [15]	36.943	7.432×10^3	1.275×10^3
	Lucy Richardson [17]	48.241	7.825×10^3	1.111×10^3
	Proposed Method	38.540	3.437×10^3	8.224×10^2

TABLE V
SUBJECTIVE TEST RESULTS OF MCIF, LR, AND THE PROPOSED METHOD

		average score (μ)	std dev (σ)
Stockholm	MCIF	0.818	1.055
	Lucy Richardson	1.54	0.415
	Proposed	1.86	0.674
Shield	MCIF	0.845	0.884
	Lucy Richardson	0.954	0.723
	Proposed	1.091	0.701

test, 11 human viewers were invited to compare the MCIF, Lucy–Richardson (LR) algorithm, and the proposed method on the picture quality improvement of Stockholm and Shield sequences. For each test, viewers were asked to compare the original and the processed sequences on separate sides of the screen. Viewers then gave a score on a continuous scale to indicate whether one image was "much better," "better," "slightly better," or "the same" as the other image. We used a 24-inch Samsung 730B LCD with 8-ms response time.

Table V shows the average and standard deviation of the subjective test scores. In the table, the average scores are all positive, meaning that the method improves the perceptual quality when compared to the original sequence. Additionally, magnitude of the average scores using the proposed method is the highest among the three methods, which implies that viewers ranked the proposed method as the best result among the three methods.

In order to test the statistical significance of the perceptual testing results, we employ the students t-test, where the null hypothesis is that the average score is $\mu = 0$, i.e., the proposed algorithm has no positive effect over the original sequence. If we let the confidence interval $\alpha = 0.01$, then the rejection region is $\mu \geq 2.359\sigma/\sqrt{N}$, where μ is the average score, σ is the standard deviation, and N is the number of viewers. It can be shown that the value $2.359\sigma/\sqrt{N}$ of MCIF, LR, and the proposed method are 0.6406, 0.2528, and 0.4105, respectively, for Stockholm, and 0.5384, 0.4404, and 0.4270, respectively, for Shield. Since all μ are greater than these figures, we conclude that all three methods give improvements to the original sequence. In addition, it can be shown that for the proposed method, the gap between the average score and the lower bound is larger than that of the other two methods. This implies that statistically the proposed method gives a more positive effect to the original sequence than the other two methods.

V. CONCLUSION

This paper has three contributions. First, we proved the equivalence between temporal and spatial integration. The equivalence allows us to simulate the LCD blur efficiently in the spatial domain, instead of a time-consuming integration in the temporal domain. Experiments verified that computing the LCD motion blur in the spatial domain is as accurate as computing it in the temporal domain. Second, we studied the limit of eye movement speed. Based on a number of papers in the cognitive science literature, we showed that perceptual quality reduces as picture motion increases. Beyond certain speed limit, human eyes cannot retrieve any useful content from the picture. Consequently, we showed that the size of the LCD motion blur filter should be limited, and the optimal size can be determined using a visual subjective test. Third, we proposed an optimization framework to preprocess the LCD signal so that it can compensate the motion blur. In order to maintain the spatial and temporal consistencies, we introduced an l_1-norm regularization function on the directional derivatives and an l_2-norm regularization function on difference between current and previous solutions. Experimental results showed that our proposed method has relatively higher PSNR, and lower spatial and temporal error than state-of-art algorithms. Future research directions include the robustness of the algorithm toward the errors introduced by motion estimation algorithms, and methods to restore texture areas.

REFERENCES

[1] S.-T. Wu and D.-K. Yang, *Fundamentals of Liquid Crystal Devices*. New York: Wiley, Sep. 2006.

[2] E. Reinhard, E. A. Khan, A. O. Akyuz, and G. Johnson*, Color Imaging Fundamentals and Applications*. Natick, MA: A. K. Peters, 2008.

[3] N. Fisekovic, T. Nauta, H. Cornelissen, and J. Bruinink, "Improved motion-picture quality of AM-LCDs using scanning backlight," in *Proc. Int. Display Worshops*, 2001, pp. 1637–1640.

[4] M. Becker, "LCD response time evaluation in the presence of backlight modulations," in *SID Symp. Tech. Dig. Papers*, 2008, vol. 39, no. 1, pp. 24–27.

[5] K. Oka and Y. Enami, "Moving picture response time (MPRT) measurement system," in *SID Symp. Tech. Dig. Papers*, 2004, vol. 35, pp. 1266–1269.

[6] J.-K. Yoon, K.-D. Kim, N.-Y. Kong, H.-C. Kim, T.-H. You, S.-S. Jung, G.-W. Han, M. Lim, H.-H. Shin, and I.-J. Chung, "LCD TV comparable to CRT TV in moving image quality—Worlds best MPRT LCD TV," in *Proc. SPIE-IS&T Electron. Imag.*, 2007, vol. 6493, p. 64930E.

[7] H.-X. Zhao, M.-L. Chao, and F.-C. Ni, "Overdrive LUT optimization for LCD by box motion blur measurement and gamma-based thresholding method," in *SID Symp. Tech. Dig. Papers*, 2008, vol. 39, no. 1, pp. 117–120.

[8] H. Wang, T. X. Wu, X. Zhu, and S.-T. Wu, "Correlations between liquid crystal director reorientation and optical response time of a homeotropic cell," *J. Appl. Phys.*, vol. 95, no. 10, pp. 5502–5508, 2004.

[9] S. Hong, B. Berkeley, and S. S. Kim, "Motion image enhancement of LCDs," in *Proc. IEEE Int. Conf. Image Process.*, 2005, pp. 11–20.

[10] B. W. Lee, K. Song, D. J. Park, Y. Yang, U. Min, S. Hong, C. Park, M. Hong, and K. Chung, "Mastering the moving image: Refreshing TFT-LCDs at 120 Hz," in *SID Symp. Tech. Dig. Papers*, 2005, pp. 1583–1585.

[11] N. Mishima and G. Itoh, "Novel frame interpolation method for hold-type displays," in *Proc. Int. Conf. Image Process.*, 2004, vol. 3, pp. 1473–1476.

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

[13] S.-J. Kang, K.-R. Cho, and Y. H. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1759–1767, Nov. 2007.

[14] H. Chen, S.-S. Kim, S.-H. Lee, O.-J. Kwon, and J.-H. Sung, "Nonlinearity compensated smooth frame insertion for motion-blur reduction in LCD," in *Proc. 7th IEEE Workshop Multimedia Signal Process.*, Nov. 2005, pp. 1–4.

[15] M. Klompenhouwer and L. Velthoven, "Motion blur reduction for liquid crystal displays: Motion compensated inverse filtering," in *Proc. SPIE-IS&T Electron. Imag.*, San Jose, 2004, p. 690.

[16] F. H. Heesch and M. A. Klompenhouwer, "Spatio-temporal frequency analysis of motion blur reduction on LCDs," in *Proc. Int. Conf. Image Process.*, 2007, vol. 4, pp. 401–404.

[17] S. Har-Noy and T. Q. Nguyen, "LCD motion blur reduction: A signal processing approach," *IEEE Trans. Image Process.*, vol. 17, no. 2, pp. 117–125, Feb. 2008.

[18] Y. Wexler, E. Shechtman, and E. Shechtman, "Space-time completion of video," *IEEE Trans. Pattern Recognit. Mach. Intell.*, vol. 29, no. 3, pp. 1–14, Mar. 2007.

[19] H. Pan, X.-F. Feng, and S. Daly, "LCD motion blur modeling and analysis," in *Proc. IEEE Int. Conf. Image Process.*, 2005, pp. 21–24.

[20] H. He, L. J. Velthoven, E. Bellers, and J. G. Janssen, "Analysis and implementation of motion compensated inverse filtering for reducing motion blur on LCD panel," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2007, pp. 1–2.

[21] T. Kurita, "Moving picture quality improvement for hold-type AM-LCDs," in *SID Symp. Tech. Dig. Papers*, 2001, pp. 986–989.

[22] S. Tourancheau, K. Brunnstrm, B. Andrn, and P. Le Callet, "LCD motion-blur estimation using different measurement methods," *J. Soc. Inf. Display*, vol. 17, no. 3, pp. 239–249, Mar. 2009.

[23] M. A. Klompenhouwer, "Temporal impulse response and bandwidth of displays in relation to motion blur," in *SID Symp. Tech. Dig. Papers*, May 2005, vol. 36, pp. 1578–1581.

[24] M. A. Klompenhouwer, "Comparison of LCD motion blur reduction methods using temporal impulse response and MPRT," in *SID Symp. Tech. Dig. Papers*, 2006, vol. 37, pp. 1700–1703.

[25] S. Yao, G. Feng, X. Lin, K. P. Lim, and W. Lin, "A coding artifacts removal algorithm based on spatial and temporal regularization," in *Proc. IEEE Int. Conf. Image Process.*, 2003, vol. 2, pp. 215–218.

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

[27] Y. Kim, K.-S. Choi, J.-Y. Pyun, B.-T. Choi, and S.-J. Ko, "A novel de-interlacing technique using bi-directional motion estimation," in *Computational Science and Its Applications*. Berlin, Germany: Springer-Verlag, 2003, pp. 957–966.

[28] S. Chan, D. Vo, and T. Nguyen, "Subpixel motion estimation without interpolation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 722–725.

[29] X. Feng, H. Pan, and S. Daly, "Comparison of motion blur measurement in LCD," in *SID Symp. Tech. Dig. Papers*, May 2007, vol. 38, no. 1, pp. 1126–1129.

[30] K. Rayner, "Eye movements in reading and information processing: 20 years of research," *Psychol. Bull.*, vol. 124, no. 3, pp. 372–422, 1998.

[31] E. Matin, "Saccadic suppression: A review," *Psychol. Bull.*, vol. 81, pp. 899–917, 1974.

[32] W. R. Uttal and E. Smith, "Recognition of alphabetic characters during voluntary eye movements," *Percept. Psychophys.*, vol. 3, pp. 257–264, 1968.

[33] J. Westerink and K. Teunissen, "Perceived sharpness in complex moving images," *Display*, vol. 16, no. 2, pp. 89–97, 1995.

[34] D. Burr, "Motion smear," *Nature*, vol. 284, no. 13, pp. 164–165, 1980.

[35] T. Bonse, "Visually adapted temporal subsampling of motion information," *Signal Process.: Image Commun.*, vol. 6, pp. 253–266, 1994.

[36] J. W. Miller and E. Ludvigh, "The effect of relative motion on visual acuity," *Surv. Ophthalmol.*, vol. 7, pp. 83–116, 1962.

[37] W. Glenn and K. Glenn, "Discrimination of sharpness in a televised moving image," *Displays*, vol. 6, pp. 202–206, 1985.

[38] K. R. Gegenfurtner, D. Xing, V. Scott, and M. Hawken, "A comparison of pursuit eye movement and perceptual performance in speed discrimination," *J. Vis.*, vol. 3, pp. 865–876, 2003.

[39] S. Chan and T. Nguyen, "Fast LCD motion deblurring by decimation and optimization," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2009, pp. 1201–1204.

[40] C. C. Paige and M. A. 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.

[41] Studies Toward the Unification of Picture Assessment Methodology Int. Telecommun. Union, Geneva, Switzerland, Rep. 1082-1, 1986, Tech. Rep., ITU.

[42] R. C. Gonzalez and R. E. Woods*, Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2007.

[43] 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.

[44] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *Int. J. Imag. Syst. Technol.*, vol. 14, no. 2, pp. 47–57, 2004.

[45] 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.

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

[47] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," *EURASIP J. Adv. Signal Process.*, vol. 2007, pp. 74585-1–74585-16, 2007.

[48] J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira, "Total variation-based image deconvolution: A majorization-minimization approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2006, vol. 2, pp. 861–864.

[49] B. Kim, "Numerical optimization methods for image restoration," Ph.D. thesis, Dept. Management Sci. Eng., Stanford Univ., Stanford, CA, Dec. 2002.

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

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

[52] Y. Wang, J. Yang, W. Yin, and Y. Zhang, An Efficient TVL1 Algorithm for Deblurring Multichannel Images Corrupted by Impulsive Noise Rice University, Houston, TX, Tech. Rep. TR-0812, Sep. 2008, CAAM.

[53] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.

[54] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[55] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Programm.*, vol. 103, pp. 127–152, 2005.

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

[57] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," Classnote of EE 392O Stanford University. Stanford, CA, Oct. 2003 [Online]. Available: http://www.stanford.edu/class/ee392o/subgrad method.pdf

[58] D. Bertsekas*, Constrained Optimization and Lagrange Multiplier Methods*. New York: Academic, 1982.

[59] N. Z. Shor*, Minimization Methods for Non-Differentiable Functions*, ser. Springer Series in Computational Mathematics. New York: Springer-Verlag, 1985.

[60] P. Gill, W. Murray, and M. Wright, *Practical Optimization*. New York: Academic, 1981.

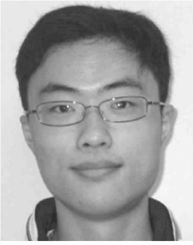
[61] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. New York: Springer-Verlag, 2000.

[62] N. Nguyen, P. Milanfar, and G. H. Golub, "A computationally efficient image superresolution algorithm," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 573–583, Apr. 2001.

[63] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*. Philadelphia, PA: SIAM, 1998.

[64] X. Zhu and P. Milanfar, "A no-reference sharpness metric sensitive to blur and noise," in *Proc. 1st Int. Workshop Quality Multimedia Experience*, July 2009 [Online]. Available: http://users.soe.ucsc.edu/~milanfar/publications/conf/qomex_zhu.pdf

[65] Methodology for the Subjective Assessment of the Quality of Television Pictures Int. Telecommun. Union, Geneva, Switzerland, BT.500-11, 2002, Tech. Rep., ITU.



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

His research interests include large-scale numerical optimization algorithms with applications to video processing.

Mr. Chan is a recipient of the Croucher Foundation Scholarship.

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

He is currently a Professor at the Department of Electrical and Computer Engineering, University of California, San Diego. He is the coauthor (with Prof. G. Strang) of a popular textbook Wavelets and Filter Banks (Wellesley-Cambridge, 1997), and the author of several MATLAB-based toolboxes on image compression, electrocardiogram compression, and filter bank design. He has over 300 publications. His research interests include video processing algorithms and their efficient implementation.

Prof. Nguyen received the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (image and multidimensional processing area) for the paper he coauthored with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks (1992), and the National Science Foundation Career Award in 1995. He was Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1994–1996), for the Signal Processing Letters (2001–2003), for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING(1996–1997, 2001–2004), and for the IEEE TRANSACTIONS ON IMAGE PROCESSING (2004–2005). He is currently the Series Editor (*Digital Signal Processing*) for Academic Press.