Virtual Focus and Depth Estimation From Defocused Video Sequences

Junlan Yang, *Student Member, IEEE*, and Dan Schonfeld, *Fellow, IEEE*

***Abstract*—In this paper, we present a novel method for virtual focus and object depth estimation from defocused video captured by a moving camera. We use the term virtual focus to refer to a new approach for producing in-focus image sequences by processing blurred videos captured by out-of-focus cameras. Our method relies on the concept of Depth-from-Defocus (DFD) for virtual focus estimation. However, the proposed approach overcomes limitations of DFD by reformulating the problem in a moving-camera scenario. We introduce the interframe image motion model, from which the relationship between the camera motion and blur characteristics can be formed. This relationship subsequently leads to a new method for blur estimation. We finally rely on the blur estimation to develop the proposed technique for object depth estimation and focused video reconstruction. The proposed approach can be utilized to correct out-of-focus video sequences and can potentially replace the expensive apparatus required for auto-focus adjustments currently employed in many camera devices. The performance of the proposed algorithm is demonstrated through error analysis and computer simulated experiments.**

***Index Terms*—Depth estimation, depth-from-defocus, image reconstruction, virtual focus.**

I. Introduction

IMAGE focus is one of the main concerns in both camera design and automated machine vision applications. Current auto-focus solutions used in commercial cameras are designed to ensure that captured images are in focus by adjusting the lens' position. A motor is used to move the position of the camera lens along the optical axis to take multiple pictures. Optimization of focus measures is subsequently used to search for the in-focus setting which is used to capture the focused image. Many image focus measures have been investigated and compared, such as gradient, Laplacian and other image moments [1]–[3]. A disadvantage of the auto-focus solution is that it requires a focal-length changing lens and an accurate engine to move the lens with a particular step size. Moreover, it has the fundamental limitation that when the scene contains multiple objects with largely varying depths, a single image cannot capture all the objects in focus simultaneously. In this paper, we propose to rely on image processing solutions, which we refer to as virtual focus. Specifically, we aim to produce images that are focused on multiple objects from out-of-focus moving cameras that have fixed lenses and no motors to move the lenses. We model the out-of-focus lens as a linear filter whose impulse response is known as the Point Spread Function (PSF). Knowledge of the PSF can be used to recover the focused image through a deconvolution process. Two major questions arise in a virtual focus problem: (1) How should we model the out-of-focus blur represented by the PSF? (2) How should we estimate the focused image using the PSF?

The use of multiple images in video sequences to improve the performance of image processing tasks has been demonstrated to be an effective approach. In particular, several algorithms have proposed to use images focused at different depths as an effective means to image reconstruction. For example, Kubota and Aizawa [5] proposed to use two images, one image focused on the foreground and the other focused on the background, in order to estimate the blur radius. A more general multiframe reconstruction algorithm has been developed for super-resolution [6]. It extends the classical single-image deconvolution methods, such as the Wiener filter, Least-Square (LS) and Maximum Likelihood (ML) estimations, to their counterparts using multiple observed images. However, super-resolution assumes a known PSF, and, thus, it addresses only the second question of estimating the focused image using the PSF. It does not solve the first question of modelling and estimating the blur represented by the PSF based on multiple images. An multiframe algorithm of out-of-focus blur estimation which has attracted a great deal of attention during the past decade is depth-from-defocus (DFD). Not only can DFD be used to estimate the blur, it also provides an estimate of the object depth, which is conventionally achieved using stereo vision [7]. The virtual focus estimation technique proposed in this paper lies in the category of DFD algorithms and provides a major contribution to PSF estimation.

The overall philosophy of DFD is based on a fundamental observation that the blur characteristic relates only to the object depth and the physical parameters of the camera system. Although this relationship is generally derived under first-order optics approximations, the observation has been verified by the reliable and consistent performance of DFD algorithms [8]. The DFD technique [9], [10] applies two settings of the camera parameters in order to acquire two images with different blur. By assuming Gaussian PSF model, a closed-form solution of the blur parameters can be derived. An extended solution has been

Manuscript received August 11, 2008; revised September 29, 2009. First published November 20, 2009; current version published February 18, 2010. This work was supported by the Physical and Digital Realization Research Center of Excellence, Motorola Labs. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mario A. T. Figueiredo.

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607-7053 USA (e-mail: jyang24@uic.edu; dans@uic.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.2009.2036708

provided in [11] considering the PSF of a cylindrical form. The PSF has also been approximated by a parametric polynomial whose coefficients can be estimated using the least-square criteria [12]. More general DFD techniques have been proposed in [13] and [14], where specific PSF models are not required. Subbarao, *et al.* [15] proposed an image recovery method under the DFD framework. In a more recent work [16], the DFD technique has been combined with stereo for image reconstruction using a Markov random field model to improve the accuracy. Studies have also been conducted in order to select the optimal camera parameters to complement DFD. The optimization criteria employed are based on minimizing the Mean-Square-Error (MSE) due to perturbations [17] or predicting the Cramer-Rao bound (CRB) [18].

Existing DFD techniques of estimating the PSF have a solid theoretical foundation, however, they impose a high requirement on the hardware. Changing camera settings, such as camera aperture and focal length, cannot be performed without a sophisticated lens system. The practical utility of DFD techniques is, therefore, limited for many imaging applications. The algorithm proposed in this paper, on the other hand, is designed for a "rigid" camera whose physical parameters are fixed. Instead of changing the camera settings to acquire multiple images with different levels of image blur, we assume a moving camera, and, thus, the position of the camera is different for distinct images. Based on the concept of DFD, the relationship between the camera motion and blur characteristics can be derived. Estimate of the blur and object depths can also be developed. Therefore, the proposed technique can be applied to inexpensive digital cameras that do not require sophisticated hardware, such as mobile-phone and web cameras. Another contribution of the proposed algorithm is to exploit multiple images in the video sequence captured by the moving camera. Images captured from different camera positions not only provide multiple images with distinct blur characteristics, but can also be used to further improve the estimation accuracy. The proposed algorithm can be used for both blur estimation and image recovery based on two or more images in the video sequence.

This paper also provides a novel method for estimation of the Phase Transfer Function (PTF). It has been a popular assumption that the PSF, and the corresponding camera lens, are spatially isotropic and, thus, its Fourier transform, also known as Optical Transfer Function (OTF), is a real-valued function with only modulus components. However, in general, this assumption does not provide an accurate model for real camera systems. The camera lens is endowed with various properties and manufacturing imperfections, which inevitably introduce a phase component to the OTF, referred to as the PTF. Modelling and estimation of the PTF has always been a challenge for image reconstruction. The difficulty in phase estimation is exacerbated by the fact that the phase often appears to be "pseudo-random" and the quality of image processing tasks is generally sensitive to the accuracy of the phase. Several techniques have been proposed for PTF estimation, including the technique of "Phase from Magnitude" based on projection onto convex sets (POCS) [19]. The main disadvantage of this approach is that it provides an iterative scheme that is not guaranteed to convergence to the true phase. In this paper, we propose a noniterative approach for estimation of the PTF within the virtual focus estimation framework.

The rest of this paper is organized as follows. Section II introduces the camera and imaging model required to define the problem of virtual focus estimation from a moving camera. In Section III, we provide three approaches to blur estimation, based on different PSF models. The noisy performance of the proposed approach is analyzed in Section IV. In Section V, we use multiple images to further improve the quality of blur estimation and video reconstruction. We extend the proposed blur estimation to incorporate the Phase Transfer Function in Section VI. Simulation results demonstrating the merit of the proposed algorithm are provided in Section VII. Finally, a summary on our results is presented in Section VIII. Preliminary results of our investigation of virtual focus from video sequences have appeared in [20] and [21].

II. Camera and Imaging Model

We begin by considering a scenario in which there is a moving camera taking a video of a static object. The camera is a rigid camera, meaning that it has a fixed lens aperture, focal length and image plane-to-lens distance. Assume one point P in the object with coordinates in camera coordinate system at time t being $[X_0, Y_0, Z_0]^T$. In time t', camera has been moved by a rotation and a translation while the point P remains in the same position in world coordinates. The new coordinates for P at time t' is $[X_1, Y_1, Z_1]^T$, also in camera coordinate system. These two coordinates can be related by the following 3-D transform:

$$[X_1 \quad Y_1 \quad Z_1]^T = R_{3\times 3} * [X_0 \quad Y_0 \quad Z_0]^T + T_{3\times 1} \tag{1}$$

where $R_{3\times 3}$, $T_{3\times 1}$ are the opposite transform of the camera's rotation and translation correspondingly. By perspective projection [22], the image coordinates of P in frame k for time t and frame k' for time t' are given by

$$[x_i \quad y_i \quad v_i]^T = \frac{\lambda_i}{Z_i} * [X_i \quad Y_i \quad Z_i]^T, \quad i = 0, 1 \tag{2}$$

where λ is the image plane-to-lens distance. From (1), (2) and by expanding $R_{3\times 3}$, $T_{3\times 1}$ to show the full entries, we have the following relationship between two image coordinates:

$$\begin{bmatrix} x_1 \\ y_1 \\ v_1 \end{bmatrix} = \frac{\lambda_1}{\lambda_0}\frac{Z_0}{Z_1} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} x_0 \\ y_0 \\ v_0 \end{bmatrix} + \frac{\lambda_1}{Z_1} \begin{bmatrix} Tx \\ Ty \\ Tz \end{bmatrix}. \tag{3}$$

Denote f to be the focal length of the camera. When point P is in-focus, we have $1/f = (1/\lambda_i) + (1/Z_i)$, $i = 0, 1$. It is easy to verify that $(\lambda_1/\lambda_0)\cdot(Z_0/Z_1) = (Z_0 - f)/(Z_1 - f)$ and $\lambda_1/Z_1 = f/(Z_1 - f)$, based on which (3) yields

$$x_1 = \frac{Z_0 - f}{Z_1 - f}(r_{11}x_0 + r_{12}y_0) + \frac{f}{Z_1 - f}(T_x + r_{13}Z_0);$$
$$y_1 = \frac{Z_0 - f}{Z_1 - f}(r_{21}x_0 + r_{22}y_0) + \frac{f}{Z_1 - f}(T_y + r_{23}Z_0).$$

The matrix form of above two equations relates $[x_0, y_0]$ in frame k and $[x_1, y_1]$ in frame k' by a 2-D affine transform:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \frac{Z_0 - f}{Z_1 - f} \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{4}$$

where $t_x = (f/(Z_1 - f))(T_x + r_{13}Z_0), t_y = (f/(Z_1 - f))(T_y + r_{23}Z_0)$. Z_0 and Z_1 are distances between the object and the camera lens, in time t and time t' respectively, which are commonly referred as depths of the object. For the sake of simplicity, we assume for the rest of the derivation that the camera observes one planar object, or the objects has one uniform depth. Under this circumstance, the parameters $(Z_0 - f)/(Z_1 - f)$, t_x and t_y become global regardless of spatial coordinates; therefore, the above affine model holds for the whole image. The assumption is also known as weak perspective projection [22], which is considered to be a good approximation when the depth variation of the object is small comparing to the field of view. In the case of large depth variation and multiple objects, the assumption that the objects in the scene have planar surfaces or near-planar surfaces is justified on local image patches whose sizes are chosen to be small enough to ensure a uniform depth. The algorithm to be proposed is then applied to the image patches instead of the entire image.

Define $s \triangleq (Z_0 - f)/(Z_1 - f)$, which represents the scaling factor between the two images as measured in terms of pixel coordinates (depths to be more specific), reflecting how the entire image scales. Denote the Fourier transform of frame k and frame k' as $F_0(u, v)$ and $F_1(u, v)$. According to the affine theorem for 2-D Fourier transform [23], (4) implies $F_0(u, v)$ and $F_1(u, v)$ have the following relationship:

$$F_1(u,v) = \frac{1}{|\Delta|} F_0\left(\frac{sr_{22}u - sr_{21}v}{\Delta}, \frac{-sr_{12}u + sr_{11}v}{\Delta}\right) \cdot exp\left\{\frac{j2\pi}{\Delta}\left[(sr_{22}t_x - sr_{12}t_y)u + (sr_{11}t_y - sr_{21}t_x)v\right]\right\} \tag{5}$$

where $\Delta \triangleq s^2(r_{11}r_{22} - r_{12}r_{21})$. Using motion estimation and image registration techniques [24], [25], we can always compensate for rotation and align the observed images prior to further processing. Therefore, we only discuss in this paper when the rotation matrix is identity, i.e., $r_{11} = r_{22} = 1$ and $r_{12} = r_{21} = 0$. It is easy to verify that (5) reduces to

$$F_1(u,v) = \frac{1}{s^2} F_0\left(\frac{u}{s}, \frac{v}{s}\right) exp\left\{j2\pi\left(t_x\frac{u}{s} + t_y\frac{v}{s}\right)\right\}. \tag{6}$$

The above derivation holds for an in-focus camera. When a camera is out of focus, the resulting image can be regarded as the in-focus image blurred by a specific PSF. A common assumption for out-of-focus PSF is that its characteristics are uniquely determined by the blur radius R. We express this blurring processing in frequency domain as the spectrum of observed blurred image $Y(u, v)$ equals the original spectrum $F(u, v)$ times the OTF $H(u, v, R)$

$$Y_i(u,v) = F_i(u,v)H(u,v,R_i), \quad i = 0, 1. \tag{7}$$

With (6) and (7), we have

$$s^2 Y_1(u,v) = Y_0\left(\frac{u}{s}, \frac{v}{s}\right) \cdot \frac{H(u,v,R_1)}{H(u/s, v/s, R_0)} \times exp\left\{j2\pi\left(u\frac{t_x}{s} + v\frac{t_y}{s}\right)\right\}. \tag{8}$$

In this and the following three sections, we model the PSF as a symmetric function, and, thus, the OTF being a real function. Therefore, we consider only the magnitude component of (8)

$$s^2 |Y_1(u,v)| = \left|Y_0\left(\frac{u}{s}, \frac{v}{s}\right)\right| \frac{H(u,v,R_1)}{H(u/s, v/s, R_0)}. \tag{9}$$

One can see that under this assumption, it is not necessary to estimate the translation parameters since the presence of translation only leads to a phase change in the frequency domain of the observed image. Therefore, estimation of the blur parameter will not be affected by translation unless the translation results in a significant change in the image content. However, as mentioned before, when in-plane rotations presented, motion estimation and registration techniques are needed to register the observed images before they can be used for estimation. In practice, the assumption of OTF being a real function may not hold in general since the imperfection of the lens system introduces phase component into OTF. We will discuss the effect of the phase component and how to estimate it in a later section.

To proceed, we need to incorporate the knowledge from optic geometry. The blur radius is given as a function of object depths and camera parameters [9]

$$R_i = \lambda L\left(\frac{1}{f} - \frac{1}{Z_i} - \frac{1}{\lambda}\right), \quad i = 0, 1 \tag{10}$$

which is equivalent to

$$Z_i = \left(\frac{1}{f} - \frac{1}{\lambda}\left(1 + \frac{R_i}{L}\right)\right)^{-1}, \quad i = 0, 1 \tag{11}$$

where λ is the image plane-to-lens distance, L is the radius of lens aperture, and Z is the depth of the object. We see that with estimates of the blur radii, the above relationship can be used to provide estimates of the object depth. It can also be seen that the blur radius is affected only by the object depth once the camera parameters are fixed. From the definition of s, we can continue to write s as a function of the blur radii R_0 and R_1

$$s = \frac{Z_0 - f}{Z_1 - f} = \frac{R_0 + L}{R_1 + L} \times \frac{R_1 + L - \lambda L/f}{R_0 + L - \lambda L/f}. \tag{12}$$

To recover the focused images from the blurred images, we need to estimate the OTF, which equals identifying the blur radius. With λ, L, f being known camera parameters, we will see in the following section that based on (9) and (12), it is possible to solve for s, R_0, R_1, thus $H(u, v, R_0)$ and $H(u, v, R_1)$.

III. Blur Estimation

In this section, we will present our algorithm based on three types of PSF. In all the cases, we begin by assuming the energy

conservation constraint [26], namely $H(0,0,R)=1$. Thus, s can be solved by noticing the DC components in (9) yields

$$s=\sqrt{Y_0(0,0)/Y_1(0,0)}. \tag{13}$$

Define $Z(u,v)$ as the ratio of two corresponding frequency components from two observations, i.e.,

$$Z(u,v)\triangleq s^2\frac{|Y_1(u,v)|}{|Y_0(u/s,v/s)|}.$$

Therefore, using (9), we obtain

$$Z(u,v)=\frac{H(u,v,R_1)}{H(u/s,v/s,R_0)}. \tag{14}$$

$Z(u,v)$ is constructed in order for the function to be determined from the observed images, which serves as the observation when estimating R_0 and R_1 from the above equation.

A. Gaussian Blur Model

A popular assumption of PSF takes form of a Gaussian function [9], in which case, we have

$$H(u,v,R)=exp\left\{-\frac{1}{4}(u^2+v^2)R^2\right\}. \tag{15}$$

Substituting (15) into (14), we can obtain

$$Z(u,v)=exp\left\{-\frac{1}{4}(u^2+v^2)\left(R_1^2-R_0^2/s^2\right)\right\}. \tag{16}$$

Note that (16) holds for all pairs of (u,v). So, an averaged solution [9] is given as follows:

$$R_1^2-\frac{R_0^2}{s^2}=\frac{1}{M_1}\sum_{(u,v)\in I_1}\frac{-4}{u^2+v^2}ln\left[Z(u,v)\right] \tag{17}$$

where I_1 is the region where the summation is well-defined, which mainly excludes frequencies where the absolute value of frequency component Y_0 has zeros values or values close to zero. M_1 is the number of (u,v) pairs in I_1. With (12), (13) and (17), we can solve for R_0 and R_1 uniquely. An approximated solution can be achieved based on the fact that $\lambda\approx f$ and $L\gg R$, under which (12) can be simplified as

$$R_1=sR_0. \tag{18}$$

It follows that $R_1^2-R_0^2/s^2=R_0^2(s^2-1/s^2)=c$. Hence, we have

$$R_0=\sqrt{\frac{cs^2}{s^4-1}}.$$

This approximation avoids measuring λ, L, f and is found to be accurate enough in experiments.

B. Geometric Blur Model

According to geometric optics, the first order approximation of the PSF for a circular lens takes the form of a cylindrical function [11]. In this case, we have

$$H(u,v,R)=2\frac{J_1(R\sqrt{u^2+v^2})}{R\sqrt{u^2+v^2}}. \tag{19}$$

Adopting the polynomial expansion [27] of a first-order Bessel function

$$J_1(x)=\frac{x}{2}-\frac{x^3}{2^2\cdot 4}+\frac{x^5}{2^2\cdot 4^2\cdot 6}-\frac{x^7}{2^2\cdot 4^2\cdot 6^2\cdot 8}+\dots$$

we have

$$H(u,v,R)=2\left(\frac{1}{2}-\frac{R^2(u^2+v^2)}{16}+\frac{R^4(u^2+v^2)^2}{384}-\dots\right).$$

Equation (14) then becomes

$$\begin{aligned}Z(u,v)&=\frac{1-R_1^2(u^2+v^2)/8+R_1^4(u^2+v^2)^2/192\dots}{1-R_0^2(u^2+v^2)/(8s^2)+R_0^4(u^2+v^2)^2/(192s^4)\dots}\\&\triangleq 1+a_1(u^2+v^2)+a_2(u^2+v^2)^2+\dots\end{aligned} \tag{20}$$

where

$$\begin{aligned}a_1&=-\frac{1}{8}\left(R_1^2-R_0^2/s^2\right)\\a_2&=\frac{1}{192}\left(R_1^4-R_0^4/s^4\right)+\frac{R_0^2}{8s^2}a_1;\dots\end{aligned} \tag{21}$$

Denote the number of coefficients to be N, i.e., $a_n, n=1\dots N$ (usually, $N=3$ gives enough accuracy). Once we identify a_n from (20), we can solve for R_0 and R_1 with (12), (13) and (21). Theoretically, identifying only a_1 is enough. However, more coefficients are desired for a reliable solution. The identification problem equals solving $[a_1,a_2,\dots,a_N]^T$ from the following matrix equation:

$$\begin{bmatrix}Z(u_0,v_0)-1\\Z(u_0,v_1)-1\\\dots\end{bmatrix}=\begin{bmatrix}u_0^2+v_0^2 & \left(u_0^2+v_0^2\right)^2 & \dots\\u_0^2+v_1^2 & \left(u_0^2+v_1^2\right)^2 & \dots\\\dots & \dots & \dots\end{bmatrix}\begin{bmatrix}a_1\\\dots\\a_N\end{bmatrix}. \tag{22}$$

Define the vector of LHS as $\mathbf{z}$, the RHS vector $\mathbf{a}\triangleq[a_1,\dots a_N]^T$. We choose $\mathbf{z}$ to contain only nonzero frequency components and assume it has a dimension of $K\times 1$. We further define the matrix on the RHS as $\mathbf{U}$, which has size of $K\times N$. Equation (22) can be written as $\mathbf{z}=\mathbf{U}\mathbf{a}$ where an least-square solution of $\mathbf{a}$ can be obtained as $\mathbf{a}=\left[\mathbf{U}^T\mathbf{U}\right]^{-1}\mathbf{U}^T\mathbf{z}$. Then we have an over-determined equation array (21) for solving R_0 and R_1.

C. Polynomial Blur Model

When we have no prior knowledge about the blur system, it becomes natural to approximate the OTF using some parametric functions. In light of the polynomial approximation for first-

order Bessel function in (20), it is reasonable to assume the two OTFs take forms of two $2M$-order polynomials

$$H(u,v,R_0) = 1 + \sum_{n=1}^{M} b_n (u^2+v^2)^n \tag{23}$$

$$H(u,v,R_1) = 1 + \sum_{n=1}^{M} c_n (u^2+v^2)^n. \tag{24}$$

In this setting, we write (14) as

$$\begin{aligned} Z(u,v) &= \frac{H(u,v,R_1)}{H(u/s,v/s,R_0)} = \frac{1+\sum_{n=1}^{M} c_n (u^2+v^2)^n}{1+\sum_{n=1}^{M} b_n (u^2+v^2)^n / s^n} \\ &\approx 1 + \sum_{n=1}^{N} a_n (u^2+v^2)^n \end{aligned}$$

where the second line is due to that $Z(u,v)$ can be approximated by a polynomial as in (20) whose coefficients can be estimated through (22). The estimation problem collapses to solve for $[b_1, b_2, \ldots b_M]^T$ and $[c_1, c_2, \ldots c_M]^T$ from $[a_1, a_2, \ldots a_N]^T$. We define

$$\mathbf{b} \equiv [b_1/s, \ldots b_n/s^n, \ldots b_M/s^M]^T, \mathbf{c} \equiv [c_1, \ldots c_n, \ldots c_M]^T$$
$$\mathbf{a}^{(1)} \equiv [a_1, \ldots, a_n, \ldots, a_M]^T, \mathbf{a}^{(2)} \equiv [a_{M+1}, \ldots, a_N]^T.$$

As long as $N \geq 2M$, a close form solution of $\mathbf{b}$ and $\mathbf{c}$ is given by [28]

$$\mathbf{c} = -\mathbf{A}^{-1}\mathbf{a}^{(2)}; \quad \mathbf{b} = \mathbf{c} - \mathbf{K}\mathbf{a}^{(1)}$$

where

$$\mathbf{A} \equiv \begin{bmatrix} a_M & \ldots & a_1 \\ a_{M+1} & \ldots & a_2 \\ \ldots & \ldots & \ldots \\ a_N & \ldots & a_M \end{bmatrix}; \ \mathbf{K} \equiv \begin{bmatrix} 1 & & & \\ c_1 & 1 & \mathbf{O} & \\ \ldots & \ldots & \ldots & \ldots \\ c_{M-1} & c_{M-2} & \ldots & 1 \end{bmatrix}.$$

After identifying $\mathbf{b}$, $\mathbf{c}$, we can continue to construct $H(u,v,R_0)$ and $H(u,v,R_1)$ for image reconstruction purposes. Nonetheless, for depth estimation purposes, we need to identify the relationship between the OTF and the blur radius. According to [12], based on the relationship between blur radius and the second moment of defocus operator, OTF has the following general property regardless of the model used for its representation:

$$\left[\frac{\partial^2 H(u,v,R)}{\partial u^2} + \frac{\partial^2 H(u,v,R)}{\partial v^2}\right]\Bigg|_{u=v=0} = -\frac{R^2}{2} \tag{25}$$

which implies

$$4b_1 = -R_0^2/2; \quad 4c_1 = -R_1^2/2. \tag{26}$$

One can verify easily that the Gaussian OTF (15) and Geometric OTF (19), as special cases, both satisfied this constraint. Based on (26), we are able to identify R_0 and R_1 from the estimated b_1 and c_1 for further depth estimation.

D. Video Reconstruction and Depth Estimation

Once we get the estimation of blur parameters for each frame, we can form the OTF for each frame individually. Each degraded frame can be passed through an inverse filter or a Wiener filter to get a reconstructed frame until the entire focused video sequence has been recovered.

Given camera physical specifications, the estimation of blur radius can also be used to obtain depth estimation for objects in the scene according to (11). In the case of 3-D scene with multiple objects, we can divide the images into small blocks and perform the depth estimation for every block to get a depth map of the scene.

IV. Noise Analysis

We discuss in this section the robustness of the proposed algorithm with respect to additive noise. We study how the estimate of OTF in the first frame changes with the disturbance of noise by introducing the noise $N(u,v)$ as an additive term in the observation model

$$Y_i(u,v) = H(u,v,R_i)F_i(u,v) + N_i(u,v); \quad i = 0,1. \tag{27}$$

For simplicity, we use the simplified relationship $R_1 = sR_0$ as in (18). Consider the Gaussian OTF. Substituting (18) into (16) and combining with (14), we have

$$s^2 \frac{|Y_1(u,v)|}{\left|Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right|} = exp\left\{-\frac{1}{4}(u^2+v^2)R_0^2(s^4-1)/s^2\right\}. \tag{28}$$

With (15) and (28), we notice that noise-free OTF estimation for first frame can be written as

$$H(u,v,R_0) = \left[s^2 \frac{|Y_1(u,v)|}{|Y_0(u/s,v/s)|}\right]^{s^2/(s^4-1)}.$$

In the case of additive noise as in (27), the estimation with presence of noises is given by

$$\hat{H}(u,v,R_0) = \left[s^2 \frac{|Y_1(u,v) - N_1(u,v)|}{|Y_0(u/s,v/s) - N_0(u/s,v/s)|}\right]^{s^2/(s^4-1)}.$$

We, thus, note the noisy estimate relates to the noise-free estimate according to the following:

$$\begin{aligned} \hat{H}(u,v,R_0) = H(u,v,R_0) \Bigg[& \frac{|Y_1(u,v)| - |N_1(u,v)|}{|Y_1(u,v)|} \\ & \times \frac{\left|Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right|}{\left|Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right| - \left|N_0\left(\frac{u}{s},\frac{v}{s}\right)\right|} \Bigg]^{\frac{s^2}{s^4-1}}. \end{aligned} \tag{29}$$

Notice that the original additive noise becomes multiplicative noise in the final estimation. The statistical characteristic of the noise has changed. If we assume the noises $N_0(u,v)$ and $N_1(u,v)$ have Gaussian distributions, then the random variable inside the square bracket is the ratio of two nonzero-mean Gaussian random variables. Its distribution has been studied in [29], based on which we can calculate the distribution and expectation of the noise. More importantly, by making a realistic assumption that the ratio between signal and noise is in general identical for two blurred images, we notice that the term inside the square bracket has value close to one. This suggests that the noisy estimate in (29) will be close to noise-free estimate, which claims the robustness of our algorithm in terms of suppressing

additive noise. This observation is confirmed by simulated experiments provided in the later experimental section.

Similar results can be obtained with other OTF models. Considering Geometric blur model to be a special case of the polynomial model, we provide briefly the result for polynomial model with a simplification that the order of polynomial is 2, i.e., $M = 1$ in (23). Combining (26) we have $H(u,v,R_i) = 1 - R_i^2(u^2+v^2)/8$, $i = 0, 1$. It is easy to show that $H(u,v,R_1) = H(u,v,sR_0) = s^2H(u,v,R_0) - s^2 + 1$. Incorporating (27) and (9), the noisy estimate can be derived after some algebra

$$\hat{H}(u,v,R_0) = (1-s^4) \times \left[\frac{|Y_1(u,v) - N_1(u,v)|}{|Y_0(u/s,v/s) - N_0(u/s,v/s)|} - s^2\right]^{-1} - s^2 + 1$$

while the noise-free estimate is given by

$$H(u,v,R_0) = (1-s^4)\left[\frac{|Y_1(u,v)|}{|Y_0(u/s,v/s)|} - s^2\right]^{-1} - s^2 + 1.$$

The relationship between the two estimates is, thus, given by

$$\frac{\hat{H}(u,v,R_0) + s^2 - 1}{H(u,v,R_0) + s^2 - 1} = \left[\frac{|Y_1(u,v)|}{\left|Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right|} - s^2\right] \times \left[\frac{|1 - N_1(u,v)/Y_1(u,v)|\,|Y_1(u,v)|}{\left|1 - N_0\left(\frac{u}{s},\frac{v}{s}\right)/Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right|\left|Y_0\left(\frac{u}{s},\frac{v}{s}\right)\right|} - s^2\right]^{-1}$$

Again by assuming the ratio of signal and noise remain identical between two images, we can conclude that the estimate with noise disturbance is close to the one without noise. These results may remain valid for higher-order polynomials; however, the derivation is much more complex.

V. Blur Estimation and Video Reconstruction Using Multiple Frames

The preceding algorithm description and analysis are presented in the context of using two images. In case of a video sequence, three or more frames are easily available. In this section, we discuss the possibility of improving the performance of our algorithm by using multiple frames. We will see that the whole system including blur estimation and image sequence reconstruction can be naturally extended to accommodate more than two input images.

A. Blur Estimation Using Multiple Frames

We can extend the blur estimation algorithm presented in Section III to incorporate multiple frames in order to improve the accuracy. Take Gaussian PSF as an example. When we have adjacent or previous (to ensure casualty) $L-1$ frames $Y_i(u,v)$, $i = 1, \ldots, L-1$, we can rewrite (16) and (18) as the following:

$$Z_i(u,v) = exp\left\{-\frac{1}{4}(u^2+v^2)\left(R_i^2 - R_0^2/s_i^2\right)\right\}$$
$$R_i/R_0 = s_i = \sqrt{Y_0(0,0)/Y_i(0,0)}$$

where $Z_i(u,v) \triangleq s_i^2(|Y_i(u,v)|/|Y_0(u/s_i, v/s_i)|)$. After some algebra, it can be shown that all the L frames can be incorporated to form one estimate for blur radius $R_0^{(0:L-1)}$

$$R_0^{(0:L-1)} = \sqrt{\frac{1}{M_2}\sum_{(u,v)\in I_2}\frac{-4}{u^2+v^2}W(u,v)}$$
$$W(u,v) \triangleq \frac{1}{L-1}\sum_{i=1}^{L-1}\frac{s_i^2}{s_i^4 - 1}ln\left[Z_i(u,v)\right] \quad (30)$$

where I_2 are defined as regions where the absolute value of the frequency spectrum at frequency component Y_0 is sufficiently large, similar as I_1 in (17). M_2 is the number of (u,v) pairs in I_2. $R_0^{(0:L-1)}$ denotes the estimate using 0 to $L-1$ frame. As we will see in the simulated experiments, the estimation based on multiple images improves largely the performance of our algorithm. It may also be useful to consider an updating scheme which updates previous estimate according to a new input frame. It can be shown that the estimate using 0 to $L-1$ frame and the estimate using 0 to L frame have the following relationship:

$$\left(R_0^{(0:L)}\right)^2 = \frac{L-1}{L}\left(R_0^{(0:L-1)}\right)^2 + \frac{1}{L}\left(R_0^{(L)}\right)^2$$

where $R_0^{(L)}$ denotes the two-frame estimate using frame L and frame 0. In other words, with a new input frame, we can perform a two-frame estimate and then update the multiple frame estimate as the weighted sum of previous multiframe estimate and current two-frame estimate. Similar extensions can be applied to geometric OTF and polynomial OTF.

B. Video Reconstruction Using Multiple Frames

In the light of multiframe blur estimation idea, it becomes natural to exploit the effects of additional frames in image reconstruction. We further find that the multiframe image reconstruction problem under our setting can be reformed as a special case of a super-resolution problem. Least-square solutions are available in frequency domain to form better estimates using multiple observed blurred images.

Denote Frame 0 as current frame to reconstruct and assume that we have adjacent or previous $L-1$ frames. Recall that (6) and (7) can be rewritten for L frames as

$$F_i(u,v) = \frac{1}{s_i^2}F_0\left(\frac{u}{s_i},\frac{v}{s_i}\right)exp\left\{j2\pi\left(t_{xi}\frac{u}{s_i} + t_{yi}\frac{v}{s_i}\right)\right\}$$
$$Y_i(u,v) = F_i(u,v)H(u,v,R_i), \quad i = 0, 1, \ldots, L-1.$$

Together they yield

$$Y_i(us_i, vs_i) = \frac{1}{s_i^2}exp\left\{j2\pi(t_{xi}u + t_{yi}v)\right\} \times H(us_i, vs_i, R_i)F_0(u,v). \quad (31)$$

Again assuming OTFs being real functions, we note $F_0(u,v)$ has the same phase component as the observed image $Y_0(u,v)$. We need only estimates of the magnitude of $F_0(u,v)$. However, in the case that OTF contains phase component, we will need to estimate the translations t_{xi} and t_{yi} before we can perform video reconstruction and in fact even prior to blur estimation.

We will detail this discussion in the next section. Take absolute values of the both side of (31), we obtain

$$|Y_i(us_i, vs_i)| = \frac{1}{s_i^2} H(us_i, vs_i, R_i) |F_0(u,v)|, \quad i = 1, \ldots, L-1.$$

Given estimates of s_i and $H(\cdot)$, we denote $Y_i'(u,v) \triangleq |Y_i(us_i, vs_i)|$ and $G_i(u,v) \triangleq (1/s_i^2) H(us_i, vs_i, R_i)$. Thus, the reconstruction problem becomes solving $|F_0(u,v)|$ from the following:

$$Y_i'(u,v) = G_i(u,v) |F_0(u,v)|, \quad i = 1, \ldots, L-1 \tag{32}$$

which can be considered as a frequency-domain expression for a super-resolution problem, except that the degradation G does not include a down-sampling. A least-square solution can be formed in frequency domain

$$\left| \hat{F}_0(u,v) \right| = \frac{\sum_{i=1}^{L-1} G_i^*(u,v) Y_i'(u,v)}{\sum_{i=1}^{L-1} |G_i(u,v)|^2}$$

which is equivalent to the spatial domain solution provided in [30] when the degradation takes form of a circulant matrix in the spatial domain (here, G^* denotes the conjugate of G). We see that multiple frames are incorporated into the reconstruction of a single frame. The solution can be further improved by introducing various regularization terms into the least-square cost function [30], which is, however, beyond the scope of this paper.

VI. Estimation of Phase Transfer Function

Section III presents our algorithm for estimating the OTF when it has no phase components, i.e., it has only its magnitude component referred as Magnitude Transfer Function (MTF). However, the real camera system does introduce a disturbance on the phase of the original image. Unfortunately, no specific knowledge on PTF is available in linear optics and it also depends significantly on physical specifications of various lenses, such as materials, shapes and sizes. Here we provide an approach for estimating the PTF under our problem setting, which works without the knowledge of physical characteristics of the lens.

The discussion will be based on two-frame for simplicity. Recall from (8) that the two blurred images have the following relationship in frequency domain:

$$s^2 \frac{Y_1(u,v)}{Y_0\left(\frac{u}{s}, \frac{v}{s}\right)} = \frac{H(u,v,R_1)}{H(u/s, v/s, R_0)} exp\left\{ j2\pi \left(u\frac{t_x}{s} + v\frac{t_y}{s} \right) \right\}. \tag{33}$$

Let us assume now the OTFs consist of MTFs $|H(u,v,R_i)|$ and PTFs $\theta_i(u,v)$

$$H(u,v,R_i) = |H(u,v,R_i)| \, exp\{ j * \theta_i(u,v) \}, \quad i = 0, 1.$$

We regard the PTF as a smooth function mainly determined by the camera system and, thus, consistent within these two frames, i.e., $\theta_1(u,v) = \theta_0(u,v) \equiv \theta(u,v)$. Therefore, a relationship on phases between the two frames can be derived from (33) as the following:

$$\angle Y_1(u,v) - \angle Y_0\left(\frac{u}{s}, \frac{v}{s}\right) - 2\pi \left(u\frac{t_x}{s} + v\frac{t_y}{s} \right) = \theta(u,v) - \theta\left(\frac{u}{s}, \frac{v}{s}\right) \tag{34}$$

where $\angle Y_i(u,v)$ denotes the phase of the images in frequency domain. In the presence of phase component, we need to estimate the translations t_x and t_y from the observed image. Motion estimation techniques proposed in [24], [25] can be implemented on the observed frames to find out the translation parameters. Note the LHS of (34) then consists of known variables. Without assuming any specific forms of θ, one idea of solving the above equation is to use Taylor expansion. We apply the 2-D Taylor expansion of the first order to $\theta(u,v)$ at the point $(u/s, v/s)$ to get

$$\theta(u,v) = \theta\left(\frac{u}{s}, \frac{v}{s}\right) + \theta_u\left(\frac{u}{s}, \frac{v}{s}\right)\left(u - \frac{u}{s}\right) + \theta_v\left(\frac{u}{s}, \frac{v}{s}\right)\left(v - \frac{v}{s}\right) \tag{35}$$

where $\theta_u(u/s, v/s)$ denotes the partial derivative of θ with respect to u evaluated at the point of $(u/s, v/s)$. Similar interpretation applies to $\theta_v(u/s, v/s)$. We denote the LHS of (34) as $C(u,v)$ and substitute (35) to (34) to obtain

$$C(u,v) = (s-1)\frac{u}{s}\theta_u\left(\frac{u}{s}, \frac{v}{s}\right) + (s-1)\frac{v}{s}\theta_v\left(\frac{u}{s}, \frac{v}{s}\right).$$

Denote $D(u/s, v/s) = C(u,v)/(s-1)$ and perform a change of variables, $\xi = u/s$ and $\eta = v/s$, we arrive at a nonlinear partial differential equation (PDE) as follows:

$$D(\xi, \eta) = \xi \frac{\partial \theta(\xi, \eta)}{\partial \xi} + \eta \frac{\partial \theta(\xi, \eta)}{\partial \eta}.$$

A general solution [31] can be obtained as follows:

$$\theta(\xi, \mu) = \int \frac{1}{\xi} D(\xi, \xi\mu) d\xi + \Phi(\mu), \quad \mu = \eta/\xi. \tag{36}$$

In the above integral, μ is considered as a fixed parameter. Φ is an arbitrary function that depends on the boundary condition of the above PDE. For simplicity, we assume $\Phi = 0$. In our case, we have discrete Fourier transforms; thus, (36) is approximately equivalent to the following discrete form:

$$\theta(\xi, \mu) = \sum_{i=0}^{\xi} \frac{1}{i} D(i, i\mu).$$

After the summation,we substitute μ by $\mu = \eta/\xi$ to get $\theta(\xi, \eta)$. And we can further obtain $\theta(u,v)$ by substituting $\xi = u/s$ and $\eta = v/s$. $H(u,v,R)$ can be constructed then by combining the estimated magnitude component and the phase component.

The multiframe image reconstruction for OTF with phase components also differs slightly from previous presentation. From (31), it is necessary to alter the definition of Y' and $G_i(u,v)$ to include phase components. Let $Y_i'(u,v) \triangleq Y_i(us_i, vs_i)$ and $G_i(u,v) \triangleq$

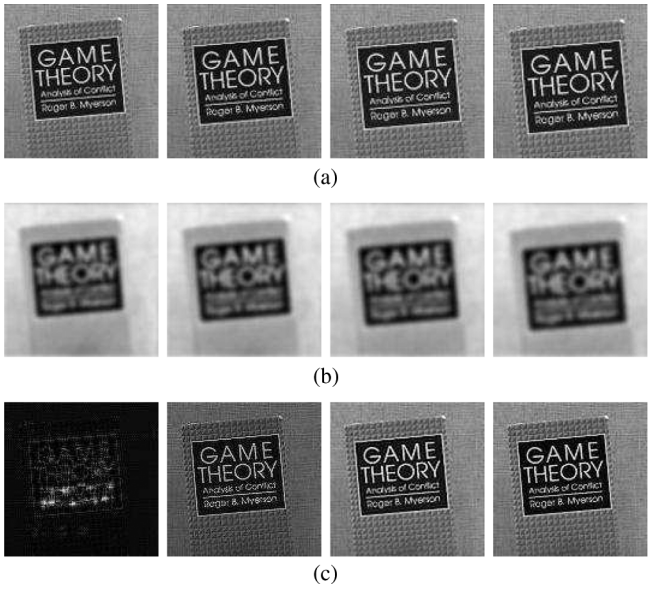

Fig. 1. Comparison between the original video sequence (a) *BOOK*, video blurred by a synthetic Geometric blur model (b) and the reconstructed first frame (c) using increasing number of frames. (a) Original video frames: Frame 1, 3, 5, and 7; (b) blurred video frames: Frame 1, 3, 5, and 7; (c) reconstructed Frame 1 using increasing number of frames.

$(1/s_i^2)exp\{j2\pi(t_{xi}u + t_{yi}v)\}H(us_i, vs_i, R_i)$ and then the same (32) can be arrived.

VII. Simulation Results

We test the effectiveness of our algorithm with synthetically blurred video sequences as well as real blur sequences. For synthetic blur tests, four video sequences are captured by a digital camcorder. The original sequences are captured in a frame rate of 10 fps and with a resolution of 320 × 240. The camcorder is mounted on a stable platform and its motion is along the optical axis, i.e., no rotations are presented. The original sequences are considered to be blur-free since the digital camcorder is set in auto-focus mode. Moreover, in all sequences, the camcorder moves towards the objects (unless otherwise notified) within a depth range between 500 and 2000 mm so that the auto-focus function of the camcorder works properly. The camera moves with a relatively low speed of approximately 30 mm per second so that motion blur has not been introduced. The groundtruth object depth for the starting frame (maximum depth) and ending frame (minimum depth) are measured and the groundtruth for each frame are given by linear interpolation due to the constant speed. Synthetic blur radii (in pixel unit) are computed according to the groundtruth depth as in (10) with $f = 22$ mm, $\lambda = 22.2$ mm and $L = 240$ pixels.

A. Planar Object and Intensity Images, Synthetic Geometric Blur

Fig. 1(a) shows the Frames 1, 3, 5, and 7 of the video sequence *BOOK*, in which the planar object is a book positioning perpendicular to the camera. Fig. 1(b) shows the sequence blurred by a simulated Geometric blur as in (19) with blur radius for Frame 1 as $R_1 = 8$ (in pixel). We perform our algorithm with the Geometric blur assumption [approximate by polynomial $\mathrm{N} = 3$ in (20)] and try to estimate the blur radii and reconstruct focused image. Fig. 1(c) shows a series of reconstructed image for Frame 1 using increasing number of frames, i.e., the second image is the reconstruction of Frame 1 computed using Frame 1, Frame 2 (not shown), and Frame 3 in the blurred sequence (b). We see that the estimation improves with the increment of number of frames, demonstrating the value of multiple-frame estimation. The estimated frames become very close to the original frames when the number of frames used exceeds 5. This is further demonstrated in Fig. 2, which shows the blur radius estimation versus groundtruth. The solid black line represents the true blur radius for Frame 1 $R_1 = 8$, with two dashed black line representing $\pm 5\% R_1$ within which the reconstruction has reasonable quality. The blue line represents the estimation of blur radius using two frames: only Frame 1 and the current frame; while the red line is the result using current frame and all the previous frames, as in (30). As we can see, although the two-frame estimations varies from $-25\% R_1$ to $25\% R_1$ among different frames used, multiple frame estimation gives steady results within $\pm 5\% R_1$ after frame number exceeds 6.

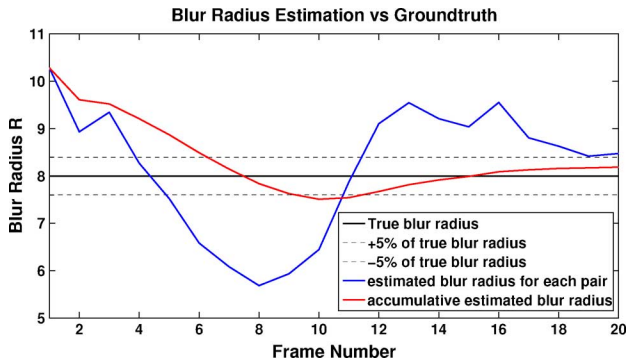

Fig. 2. Comparison of estimated blur radius for Frame 1 in video sequence *BOOK* using two-frame (blue line) and multiframe (red line); x-axis represents the number of frames used for multiframe estimation and the frame index for two-frame estimation correspondingly. The solid black line represents the true blur radius for Frame 1 $R_1 = 8$, with two dashed black line representing $\pm 5\% R_1$.

B. Approximately Planar Object and Color Images, Synthetic Gaussian Blur

Fig. 3(a) shows the Frames 1, 101, 161, and 206 of a long video sequence *SOCCER*, in which the object soccer is an approximately planar object. The camera moves from 1300 mm towards the object and moves backwards after it reaches the distance of 800 mm. This video sequence consists of color images. Although previous discussions only consider intensity images, the whole virtual focusing system can be extended to deal with color images by simply applying the algorithm to each of the RGB color components. Fig. 3(b) shows the sequence blurred by a simulated Gaussian blur as in (15). Fig. 3(c) shows the reconstructed sequence where 5 frames are used for estimating each frame. The number of frames used is selected to ensure that we have sufficient frames, i.e., the resulting reconstruction quality is stable. As can be seen, the estimation of focused sequence gives constantly good performance. We also present the depth estimation result in Fig. 4. The solid black line represents the true depth, with two dashed black lines representing ±5% of the truth depth. The blue line represents the estimation of depth

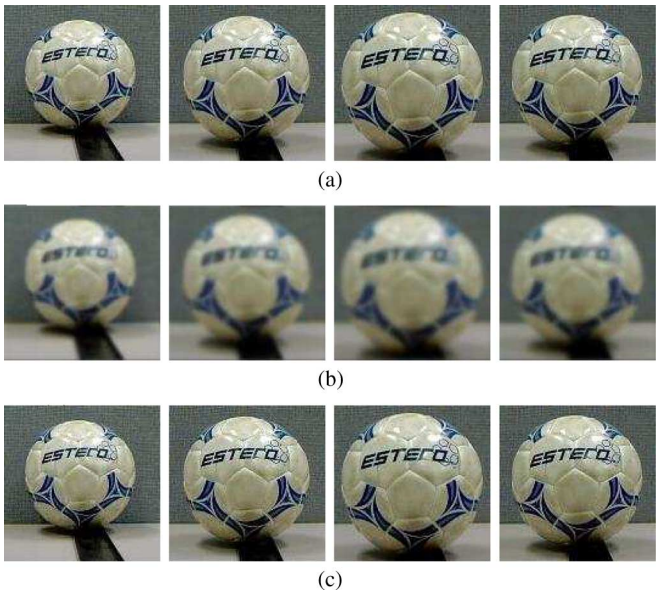

Fig. 3. Comparison between the original video sequence (a) *SOCCER*, video blurred by a synthetic Gaussian blur model (b) and the reconstructed video sequence (c) when using 5 frames for estimating each blur radius. (a) Original video frames: Frames 1, 101, 161, and 206; (b) blurred video frames: Frames 1, 101, 161, and 206; (c) reconstructed video frames: Frames 1, 101, 161, and 206.

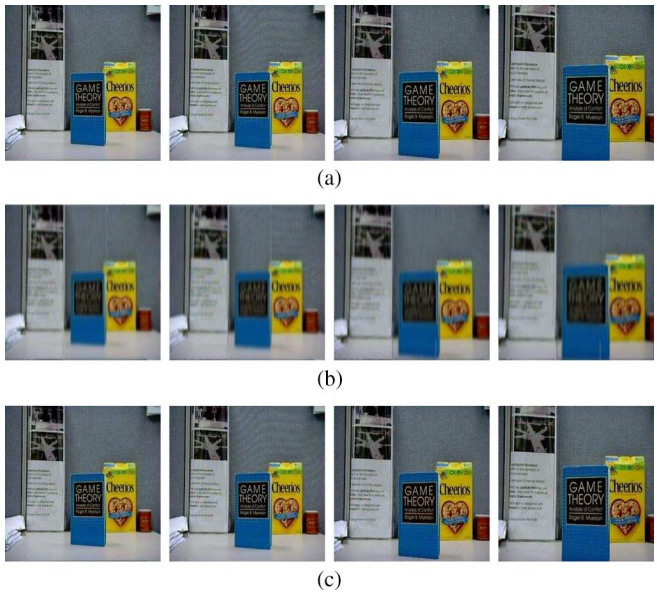

Fig. 5. Comparison between the original video sequence (a) *DESK*, video blurred by a synthetic Gaussian blur model (b) and the reconstructed video sequence (c) when using 10 frames for estimating each blur radius. (a) Original video frames: Frames 31, 61, 91, and 121; (b) blurred video frames: Frames 31, 61, 91, and 121; (c) reconstructed video frames: Frames 31, 61, 91, and 121.

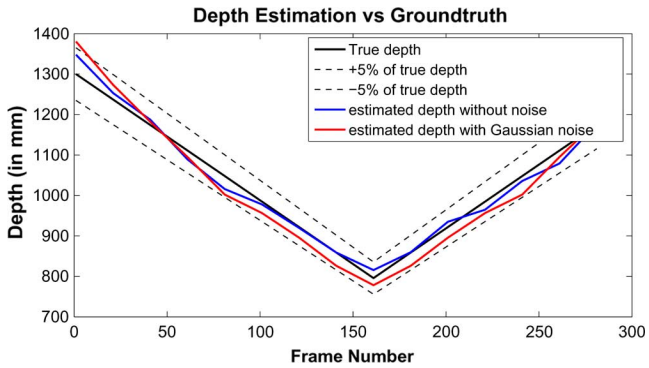

Fig. 4. Estimated depth (blue line) for video sequence *SOCCER* comparing to the groundtruth as well as the estimated depth when the observed images are contaminated by an additive Gaussian noise (red line); x-axis represents the the frame index. The solid black line represents the true depth, with two dashed black lines representing $\pm 5\%$ of the groundtruth.

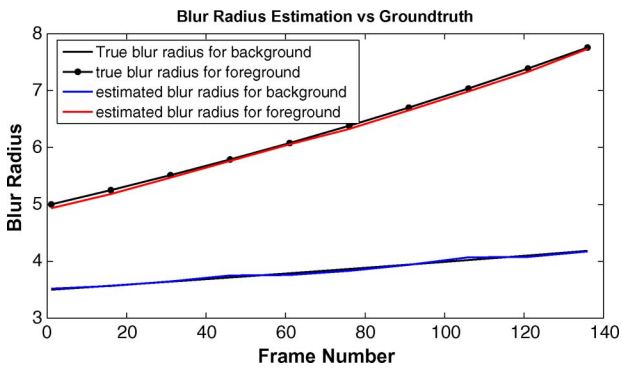

Fig. 6. Estimated blur radius for background (blue line) and foreground (red line) in video sequence *DESK* comparing to groundtruths (black lines).

using the proposed blur estimation and resulting depth estimation. We see that the depth estimation is within the $\pm 5\%$ of the groundtruth throughout the whole sequence.

Moreover, we test the robustness of the proposed algorithm by adding a zero-mean Gaussian noise (standard deviation of 10 pixel values, with all images having 265 graylevels) to the blurred image sequence. The depth estimation result is illustrated as the red line in Fig. 4. We can see that although the presence of noise has degraded the accuracy of the estimation, depth estimates with this moderately large noise still lie within the $\pm 5\%$ of the groundtruth. We conclude that the proposed algorithm has plausible robustness with respect to additive noise as we have shown theoretically in Section IV.

C. Three-Dimensional Scene and Color Images, Synthetic Gaussian Blur

Fig. 5(a) shows the Frames 31, 61, 91, and 121 of a color video sequence *DESK*, where multiple objects form a background-foreground scene. Gaussian synthetic blurs are added according to the depths of the objects as shown in Fig. 5(b). As mentioned in Section III, our algorithm can be applied to local regions of the image to ensure each region has the same depth. In a rather simple case as in *DESK*, we can divide the frames into background regions and foreground regions. Fig. 5(c) shows the reconstructed sequence where we use 10 frames for estimating each frame. It can be seen that our algorithm gives high-quality reconstruction for both the foreground objects and the background. Fig. 6 is a plot with the groundtruth blur radius for both the background (solid black line) and foreground (black line with dots). The simulated blur increases when the depth decreases. The blue line and the red line represent the blur estimation for background and foreground respectively. They are both close to the groundtruth, which verifies the competence of our algorithm in dealing with 3-D scenes.

D. Reconstruction With PTF Estimation, Synthetic Blur

We test our PTF estimation with a sequence *ALARM* as shown in Fig. 7; (a) shows the first frame and (b) shows the blurred first

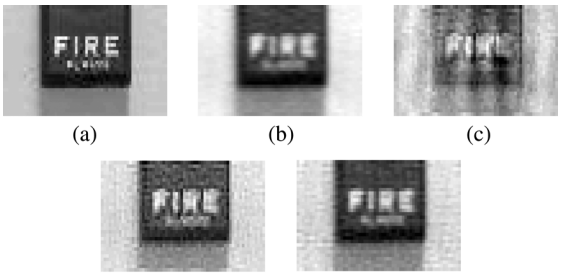

Fig. 7. Comparison of focused image reconstruction methods for video ALARM: (a) original image; (b) blurred image, $\mathrm{PSNR} = 18.47$ dB; (c) focused image reconstruction using restoration from magnitude (RFM), $\mathrm{PSNR} = 15.62$ dB; (d) focused image reconstruction using blur function magnitude estimation, $\mathrm{PSNR} = 18.99$ dB; and (e) focused image reconstruction using blur function magnitude and phase estimation, $\mathrm{PSNR} = 21.65$ dB.

frame as a result of an synthetic OTF consisting of a Gaussian MTF and an arbitrary PTF. Fig. 7(c) shows the reconstruction result using Restoration from Magnitude (RFM) in [19]. It is a technique based on projection onto convex set (POCS) while the two convex sets are the set of space-limited functions and the set of all functions that have a Fourier transform magnitude equal to a prescribed function. Fig. 7(d) shows the reconstruction result using only proposed OTF magnitude estimation and PTF is considered to be zero. Fig. 7(e) shows the reconstruction result using both proposed OTF magnitude estimation and phase estimation. It can be seen that the restoration including PTF estimation performs better than the restoration without phase and restoration using RFM, which verifies the effectiveness of our PTF estimation algorithm. It is more explicitly demonstrated through PSNR improvements. The blurred image has a PSNR of 18.47 dB while the reconstruction using RFM has a PSNR of 15.62 dB. The reconstruction using proposed magnitude-only OTF estimation gives a slightly improved PSNR of 18.99 dB. The reconstruction using both magnitude and phase estimation gives the highest PSNR of 21.65 dB, which is an over 3-dB improvement comparing to the blurred image.

E. Real Blurred Sequence

We also test our algorithm with real blur image sequences. The sequences are captured by a web camera whose lens can be manually preadjusted, but will remain fixed during the whole capturing process. We set the lens in an out-of-focus position for a certain object in a certain depth, and take videos while moving the camera. The physical parameters f and λ are not available after adjustments; thus, the simplified camera geometry (18) will be used. The sequences are captured in a frame rate of 10 fps and with a resolution of 320×240.

Fig. 8(a) shows the Frames 20, 30, 40, and 50 of a B/W video sequence *SPIRAL*, in which the object is a cover picture of a book and placed perpendicular to the camera. The webcam moves manually towards the object, similar as in the synthetic cases except that the captured sequence contains small translations attributed to an unsteady hand during video capture. The distance between the camera and the object is ranging from 50 mm to 100 mm. We preset the lens to focus in near distance, i.e., small depth. Thus, when the camera moves forward, the captured video frames observes less blur effects. The whole video consists of 60 frames. Fig. 8(c)–(e) shows corresponding

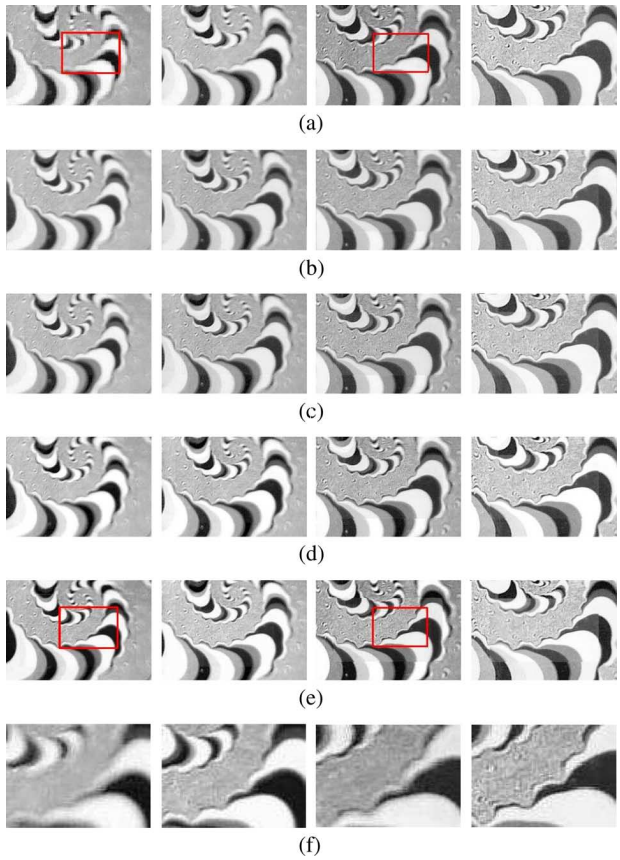

Fig. 8. Reconstruction for a real blur video *SPIRAL* using multiframe blur estimation and multiframe image reconstruction: (a) Original video frames: Frames 20, 30, 40, and 50; (b) Gaussian blur model reconstruction without phase estimation; (c) Gaussian blur model reconstruction with phase estimation; (d) geometric blur model reconstruction with phase; (e) polynomial blur model reconstruction with phase; (f) enlarged rectangles. From left to right: Original Frame 20; reconstructed Frame 20 using polynomial model; original Frame 40 and reconstructed Frame 40 using polynomial model.

reconstructed frames using the proposed multiframe blur estimation and multiframe image reconstruction. Five immediately preceding frames are used for reconstructing each frame. Motion estimation has been employed to find out the translational parameters required for PTF estimation. Since the form of the blur function is unknown, we provide reconstruction results based on the three different models for the magnitude component: Gaussian blur model [Fig. 8(c)], Geometric blur model [Fig. 8(d)] and the general polynomial model [Fig. 8(e)]. PTF estimation does not assume an OTF model and is, thus, identical for all three OTF models considered. We also provide in Fig. 8(b) the result of magnitude-only reconstruction based on Gaussian blur model, for comparison. We note that the phase disturbance is relatively small in this experiment, and, thus, the improvement in the results using PTF is only slightly superior to the results obtained without PTF estimation.

It can be seen from the results the reconstruction improves the quality of the blur sequence with sharper edges and more details. Fig. 8(f) shows the enlarged portions indicated as rectangle areas in the original frames, for clearer comparisons. All of three OTF models yield similar performance with the polynomial model being slightly better. Especially for Frame 20 and 30, polynomial model gives sharper edges. However, in the case that computational cost is a crucial constraint, the Gaussian OTF

is a more desirable model since its computational complexity is much lower than the polynomial model. When implemented in MATLAB on a PC with a single 3.2 GHz CPU, the five-frame blur estimation with Gaussian OTF assumption requires only 0.6 seconds while the Geometric model consumes 19 seconds and the polynomial model consumes 31 s. Since all the video sequences in our experiments have the same resolution, identical computational speeds are observed with videos discussed previously.

VIII. Conclusion

In this paper, we introduced a novel method for virtual focus and object depth estimation from defocused videos. The proposed algorithm exploits differences in the blur characteristics of adjacent video frames captured by an out-of-focus moving camera. Multiple frames are used to further improve the system's performance. We explored several blur models which can be used to recover arbitrary transfer functions. Analysis of the effect of noise on the proposed approach to blur estimation indicates that the algorithm performs robustly with the disturbance of noise. Computer simulated experiments confirm the merit of our approach to virtual focus estimation.

The main advantage of the proposed algorithm is that it works with a rigid lens system, while existing methods require a sophisticated apparatus for lens adjustment. It, therefore, has the potential to be deployed in cell-phone and web cameras, where the lens systems are often inexpensive and do not have a mechanism to adjust the position of the lens for auto-focus capability. Furthermore, the proposed algorithm can also be used as a post-processing technique to correct video sequences which suffer from out-of-focus blur.

We have shown in Section II that an arbitrary 3-D affine motion can be approximated by a 2-D affine motion. More complex motions can be modelled fairly accurately as affine motions on local image patches. Moreover, in the case that the 2-D affine motion contains rotation, we rely on motion compensation to correct the rotation. Therefore, with the aid of an image registration preprocessing module, the algorithm developed can deal with common motions used to model video capture in mobile cameras. Following a similar argument to [7], our estimation technique can rely on image patches containing isolated objects; thus possible object self-occlusions in the video will only introduce limited, local degradation in the system's performance.

References

[1] E. Krotkov, "Focusing," *Int. J. Comput. Vis.*, vol. 1, pp. 223–237, 1987.

[2] M. Subbarao and J. Tyan, "Selecting the optimal focus measure for autofocusing and depth-from-focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 864–870, Aug. 1998.

[3] P. T. Yap and P. Raveendran, "Image focus measure based on chebyshev moments," *IEE Proc. Vis. Image Signal Process.*, vol. 151, no. 2, pp. 128–136, 2004.

[4] Y. Zhang, Y. Zhang, and C. Wen, "A new focus measure method using moments," *Image Vis. Comput.*, vol. 18, pp. 959–965, 2000.

[5] A. Kubota and K. Aizawa, "Reconstructing arbitrarily focused images from two differently focused images using linear filters," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1848–1859, Nov. 2005.

[6] S. Borman and R. Stevenson, Spatial Resolution Enhancement of Low-Resolution Image Sequences—A Comprehensive Review With Directions for Future Research, Univ. Notre Dame, South Bend, IN, 1998, Tech. Rep..

[7] Y. Schechner and N. Kiryati, "Depth from defocus vs. stereo: How different really are they?," *Int. J. Comput. Vis.*, vol. 89, pp. 141–162, 2000.

[8] A. Pentland, T. Darrell, M. Turk, and W. Huang, "A simple, real-time range camera," presented at the Computer Vision and Pattern Recognition, San Diego, CA, Jun. 1989.

[9] M. Subbarao, "Efficient depth recovery through inverse optics," in *Machine Vision for Inspection and Measurement*, H. Freeman, Ed. New York: Academic, 1989.

[10] A. Pentland, "A new sense for depth of field," *IEEE Trans. Pattern Recongnit. Mach. Intell.*, vol. 9, pp. 523–531, 1987.

[11] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 2, pp. 97–108, Feb. 1993.

[12] J. Rayala, S. Gupta, and S. Mullick, "Estimation of depth from defocus as polynomial system identification," *IEE Proc.*, pp. 356–362, 2001.

[13] M. Subbarao and T. Wei, "Depth from defocus and rapid autofocusing: A practical approach," *Proc. Comput. Vis. Pattern Recognit.*, vol. 9, pp. 773–776, 1992.

[14] G. Surya, "Three Dimensional Scene Recovery From Image Defocus," Ph.D thesis, Dept. Elect. Eng., State Univ. New York, Stony Brook, 1994.

[15] M. Subbarao, T. Wei, and G. Surya, "Focused image recovery from two defocused images recorded with different camera settings," *IEEE Trans. Image Process.*, vol. 4, no. 12, pp. 1613–1628, Dec. 1995.

[16] A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1521–1525, Nov. 2004.

[17] Y. Schechner and N. Kiryati, "The optimal axial interval in estimating depth from defocus," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 843–848.

[18] A. Rajagopalan and S. Chaudhuri, "Optimal selection of camera parameters for recovery of depth from defocused images," *Proc. Comput. Vis. Pattern Recognit.*, vol. 18, pp. 219–224, 1997.

[19] A. Levi and H. Stark, "Image restoration by the method of generalized projections with application to restoration from magnitude," *J. Opt. Soc. Amer. A*, vol. 1, no. 9, pp. 932–943, Sep. 1984.

[20] J. Yang, D. Schonfeld, and M. Mohamed, "Robust focused image estimation from multiple images in video sequences," presented at the IEEE Int. Conf. Image Processing, San Antonio, TX, Sep. 2007.

[21] J. Yang, D. Schonfeld, and M. Mohamed, "Focused video estimation from defocused video sequences," presented at the SPIE, Vis. Commun. Image Process., San Jose, CA, Jan. 2008.

[22] D. A. Forsyth and J. Ponce, *Chapter I, Computer Vision: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

[23] R. Bracewell, K. Chang, A. Jha, and Y. Wang, "Affine theorem for two dimensional fourier transform," *Electron. Lett.*, Feb. 1993.

[24] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," presented at the IS&T/SPIE Symp. Electronic Imaging, Image and Video Communication, Santa Clara, CA, Jan. 2003.

[25] J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, "Online video stabilization based on particle filters," presented at the IEEE Int. Conf. Image Processing, Atlanta, GA, Nov. 2006.

[26] R. Bracewell, *Two-Dimensional Imaging*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[27] F. Bowman, *Introduction to Bessel Functions*. New York: Dover, 1958.

[28] D. Graupe, *Time Series Analysis, Identification and Adaptive Filtering*. Melbourne, FL: R.E. Kreiger, 1989.

[29] D. Hinckley, "On the ratio of two correlated normal random variables," *Biometrika*, 1969.

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

[31] A. D. Polyanin, V. F. Zaitsev, and A. Moussiaux, *Handbook of First Order Partial Differential Equations*. New York: Taylor and Francis, 2002.

Junlan Yang (S'06) received the B.S. degree in information engineering in 2005 from Zhejiang University, Hangzhou, China. She is currently pursuing the Ph.D. degree in Department of Electrical and Computer Engineering, University of Illinois, Chicago, IL.

She has been a research assistant in Multimedia Communications Laboratory since 2005.

Ms. Yang received the IBM Student Paper Award at the IEEE International Conference on Image Processing in 2007.

Dan Schonfeld (M'90–SM'05–F'09) received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Johns Hopkins University, Baltimore, MD, in 1986, 1988, and 1990, respectively.

He joined University of Illinois at Chicago in 1990, where he is currently a Professor in the Departments of Electrical and Computer Engineering, Computer Science, and Bioengineering, and Co-Director of the Multimedia Communications Laboratory (MCL). He has authored over 170 technical papers in various journals and conferences.