Focused Video Estimation from Defocused Video Sequences

Junlan Yang[a], Dan Schonfeld[a] and Magdi Mohamed[b]

[a]Multimedia Communications Lab, ECE Dept., University of Illinois, Chicago, IL
[b]Physical Realization Research Center of Excellence, Motorola Labs, Schaumburg, IL

ABSTRACT

This paper proposes a novel technique for estimating focused video frames captured by an out-of-focus moving camera. It relies on the idea of Depth from Defocus (DFD), however overcomes the shortage of DFD by reforming the problem in a computer vision framework. It introduces a moving-camera scenario and explores the relationship between the camera motion and the resulting blur characteristics in captured images. This knowledge leads to a successful blur estimation and focused image estimation. The performance of this algorithm is demonstrated through error analysis and computer simulated experiments.

Keywords: Depth from Defocus (DFD), Focusing, Camera motion, Image restoration

1. INTRODUCTION

Focusing is one of the persistent concerns in camera design. Current auto-focus solutions in commercial digital cameras are mainly based on focus measures. A digital camera equipped with auto-focus function moves its adjustable lens to take multiple pictures and then find the best picture by minimizing certain out-of-focus penalties. The shortage of auto-focus solution is that it requires a focal-length changing lens and an accurate lens engine to move the lens with a particular step size. From a different point of view however, image processing solutions model the out-of-focus lens as a linear filter whose impulse response is known as the Point Spread Function (PSF). The focused images can be recovered through a deconvolution process.

The overall philosophy of estimating PSF and its Fourier transform, also known as Optical Transfer Function (OTF), is based on a fundamental observation that the blur characteristic relates only to the object depth and the camera physical parameters. Despite that the relationship is usually simplified using first order optics, this observation verifies itself through the success of depth-from-defocus (DFD) algorithms. Classic DFD technique[1] applies two settings of camera parameters for acquiring two differently blurred images. Assuming a Gaussian PSF, a close form estimation of the blur parameter can be achieved.[1] A solution is provided in[2] when the PSF has a cylindrical form. More generally, OTF can be approximated by a parametric polynomial[3] and the parameters can be estimated using a least-square criteria. In a more recent work,[4] the technique of DFD is combined with stereo pairs and the estimation is performed with Markov random fields to improve the accuracy.

DFD for estimating the PSF has a solid and elegant theoretical foundation, however it poses a high requirement on the hardware. Due to the fact that changing camera settings such as camera aperture and focal length cannot be done without sophisticated lens system, it limits the applications in practice. The algorithm proposed in this paper, on the other hand, is designed for a 'rigid' camera whose physical parameters are all fixed. Therefore it can be applied to simple digital cameras especially webcams and mobile-phone cameras. Another novelty of our algorithm is to take advantage of multiple images available in the videos taken by moving cameras. Images taken in various positions not only provide multiple differently blurred images for estimation but also help to improve the estimation accuracy.

We also provide in this paper a novel idea on the estimation of Phase Transfer Function (PTF). While it is a popular assumption that the PSF being spatially symmetric and thus OTF being a real function with only its modulus component known as Modulus Transfer Function (MTF), it is in general not an accurate model for real cameras. Camera lens with various properties and manufacturing imperfections introduce inevitably the phase component to OTF referred as the PTF. The modelling and estimation of PTF has always been a challenge for image reconstruction, where several techniques have been proposed including the techniques of Phase from

E-mail:jyang24,dans@uic.edu, Magdi.Mohamed@motorola.com

Magnitude based on projection onto convex sets.[5] Common disadvantages for these approaches are their iterative schemes and no guarantee of convergence to the true phase. In this paper, we propose a non-iterative approach for the estimation of PTF based on our framework.

The rest of this paper is organized as follows. In Section 2, we begin with the problem definition. In Section 3, we explain the main idea of blur estimation through two examples of PSF and the idea of multiple-image estimation. Section 4 analyzes the noise performance for the system. Section 5 discusses the estimation of PTF. Section 6 provides the simulation results and Section 7 draws the conclusion.

2. 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. One point in the object projects onto different image coordinates when the camera moves. In time t and time $t^{'}$, the camera takes two images, frame k and frame $k^{'}$. The pixel locations (x_0, y_0) in image frame k and (x_1, y_1) in frame $k^{'}$ are related by a 2D affine transform:[6]

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \frac{z_0 - f}{z_1 - f} \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{1}$$

where $r_{11}, r_{12}, r_{21}, r_{22}$ are rotation parameters and t_x, t_y are translation parameters of the camera motion. 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 simplicity of derivation, here we assume that the camera captures only planar objects. In the case of 3D objects and 3D scenes, one can apply directly the whole algorithm to a neighborhood of each pixel in the frames, with the neighborhood size small enough to ensure a uniform depth. f is the focal length of the camera.

Define $s \equiv \frac{z_0 - f}{z_1 - f}$. 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 2D Fourier transform,[7] $F_0(u, v)$ and $F_1(u, v)$ has the following relationship:

$$F_1(u,v) = \frac{1}{|\Delta|} F_0\left(\frac{sr_{22}u - sr_{21}v}{\Delta}, \frac{-sr_{12}u + sr_{11}v}{\Delta}\right) exp\left\{\frac{j2\pi}{\Delta}[(sr_{22}t_x - sr_{12}t_y)u + (sr_{11}t_y - sr_{21}t_x)v]\right\}, \tag{2}$$

where $\Delta \equiv s^2(r_{11}r_{22} - r_{12}r_{21})$. Using the motion estimation and stabilization technique,[6] we can compensate for rotation before we further process the images. Therefore we only discuss in this paper when the rotation matrix is identity. Then (2) can be simplified as

$$F_1(u,v) = \frac{1}{s^2} F_0\left(\frac{u}{s}, \frac{v}{s}\right) exp\{j2\pi(t_x \frac{u}{s} + t_y \frac{v}{s})\}. \tag{3}$$

We model in this section the PSF to be a symmetric function and thus the OTF being a real function. Therefore we consider only the magnitude component of (3):

$$|F_1(u,v)| = \frac{1}{s^2} |F_0\left(\frac{u}{s}, \frac{v}{s}\right)|. \tag{4}$$

When a camera is out of focus, the resulting image is blurred by a specific PSF, whose parameters are uniquely determined by the blur radius R. In frequency domain, the spectrum of blurred image $Y(u, v)$ will be the original spectrum times the OTF $H(u, v, R)$:

$$Y_i(u,v) = F_i(u,v)H(u,v,R_i), \qquad i = 0, 1. \tag{5}$$

With (4) and (5), we have

$$s^2|Y_1(u,v)| = |Y_0\left(\frac{u}{s}, \frac{v}{s}\right)| \frac{H(u,v,R_1)}{H(u/s, v/s, R_0)}. \tag{6}$$

To proceed, we need to incorporate the knowledge from optic geometry. The blur radiuses are given as a function of depths and camera parameters:[1]

$$R_i = vL(\frac{1}{f} - \frac{1}{z_i} - \frac{1}{v}), \qquad i = 0, 1. \tag{7}$$

where v is the image plane-to-lens distance, L is the radius of lens aperture, and z is the depth of the object. It can be seen that the blur radius is affected only by the depth of the object for one particular camera. From the definition of s we can continue to arrive at

$$s = \frac{z_0 - f}{z_1 - f} = \frac{R_0 + L}{R_1 + L} \times \frac{R_1 + L - vL/f}{R_0 + L - vL/f}. \tag{8}$$

To estimate the focused images from the blurred images, we need to estimate the OTF, which equals identifying the blur radiuses. With v, L, f being known camera parameters, it is able to solve for s, R_0, R_1, thus $H(u, v, R_0)$ and $H(u, v, R_1)$, based on (6) and (8).

3. BLUR PARAMETER ESTIMATION AND VIDEO RECONSTRUCTION

In this section, we will present our algorithm based on two types of PSF. In both cases, we begin with assuming the energy conservation constraint, namely $H(0, 0, R) = 1$. Thus, s can be solved by noticing the DC components in (6) yields

$$s = \sqrt{Y_0(0, 0)/Y_1(0, 0)}. \tag{9}$$

3.1 Gaussian Blur Model

When PSF takes the form of a Gaussian function,[1] we have:

$$H(u, v, R) = exp\{-\frac{1}{4}(u^2 + v^2)R^2\}. \tag{10}$$

Substituting (10) to (6), we can obtain

$$s^2 \frac{|Y_1(u, v)|}{|Y_0(\frac{u}{s}, \frac{v}{s})|} = exp\{-\frac{1}{4}(u^2 + v^2)(R_1^2 - R_0^2/s^2)\}. \tag{11}$$

Equation (11) is true for all pairs of (u, v), so an averaged solution[1] is given as follows:

$$c \equiv \frac{1}{M_1} \sum_{(u,v)\in I_1} \frac{-4}{u^2 + v^2} ln(s^2 \frac{|Y_1(u, v)|}{|Y_0(\frac{u}{s}, \frac{v}{s})|}), \tag{12}$$

$$R_1^2 - R_0^2/s^2 = c, \tag{13}$$

where I_1 is the region within which the summation is well-defined and M_1 is the number of (u, v) pairs in I_1. With (8), (9) and (13), we can solve R_0 and R_1 uniquely. An approximated solution can be achieved based on the fact that $v \approx f$ and $L \gg R$. (8) can be then simplified as $R_1 = sR_0$, so that $R_1^2 - R_0^2/s^2 = R_0^2(s^2 - 1/s^2) = c$. Hence, $R_0 = \sqrt{\frac{cs^2}{s^4-1}}$. This approximation avoids measuring v, L, f and is found to be accurate enough in experiments.

3.2 Geometric Blur Model

According to geometric optics, the first order approximation of the PSF takes the form of a cylindrical function in the case of a circular aperture.[2] Therefore we have

$$H(u, v, R) = 2\frac{J_1(R\sqrt{u^2 + v^2})}{R\sqrt{u^2 + v^2}}. \tag{14}$$

Adopting the polynomial expansion[8] of a bessel function,

$$J_1(x) = \frac{x}{2} - \frac{x^3}{2^2 \cdot 4} + \frac{x^5}{2^2 \cdot 4^2 \cdot 6} - \frac{x^7}{2^2 \cdot 4^2 \cdot 6^2 \cdot 8} + ..., \tag{15}$$

Equation (6) becomes

$$\begin{aligned} s^2 \frac{|Y_1(u,v)|}{|Y_0(\frac{u}{s}, \frac{v}{s})|} &= 1 + a_1(u^2+v^2) + a_2(u^2+v^2)^2 + ..., \\ a_1 = -\frac{1}{8}(R_1^2 - R_0^2/s^2); a_2 &= \frac{1}{192}(R_1^4 - R_0^4/s^4) + \frac{R_0^2}{8s^2}a_1; ... \end{aligned} \tag{16}$$

Once we identify $a_n, n = 1...N$, we can solve for R_0 and R_1 with (8) and (9). N is the number of coefficients. Theoretically, identifying only a_1 is enough. However, more coefficients are desired for a reliable solution. The identification problem equals solving the following matrix equation:

$$\begin{bmatrix} Z(u_0,v_0) \\ Z(u_0,v_1) \\ ... \end{bmatrix} = \begin{bmatrix} 1 & u_0^2+v_0^2 & (u_0^2+v_0^2)^2 & ... \\ 1 & u_0^2+v_1^2 & (u_0^2+v_1^2)^2 & ... \\ 1 & ... & ... & ... \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ ... \\ a_N \end{bmatrix} \tag{17}$$

where $Z(u,v) \equiv s^2 \frac{|Y_1(u,v)|}{|Y_0(u/s, v/s)|}$. The LHS is a $K \times 1$ vector, where K is the number of non-zero frequency components being used. The matrix in RHS is of size $K \times N$, and the vector in RHS is the unknown vector of size $N \times 1$. An least-square solution of $[a_1, ...a_N]$ can be obtained from (17) then we will have an over-determined equation array (16) for solving R_0 and R_1.

Once we get the estimation of the transfer function, we can process the degraded image with an inverse filter or a Wiener filter to recover the focused image. And we apply this estimation technique to successive video frames until the entire focused video sequence has been recovered.

3.3 Multiple Image Estimation

When three or more frames available, we can use them together for estimation in order to improve the accuracy. For instance, with Gaussian PSF, if we have a third image $Y_2(u,v)$, we can form another set of equations using the simplified version of (8):

$$\begin{aligned} s'^2 \frac{|Y_2(u,v)|}{|Y_0(\frac{u}{s'}, \frac{v}{s'})|} &= exp\{-\frac{1}{4}(u^2+v^2)(R_2^2 - R_0^2/s'^2)\}; \\ R_2/R_0 = s' &= \sqrt{Y_0(0,0)/Y_2(0,0)}. \end{aligned} \tag{18}$$

Along with (11), (9) and $R_1 = sR_0$, we have six equations for five unknowns. It is overdetermined, which enables us to use information from three frames to form one estimation:

$$w \equiv \frac{1}{M_2} \sum_{(u,v) \in I_2} \frac{-4}{u^2+v^2} ln(s^2 \frac{|Y_1(u,v)|}{|Y_0(\frac{u}{s}, \frac{v}{s})|} s'^2 \frac{|Y_2(u,v)|}{|Y_0(\frac{u}{s'}, \frac{v}{s'})|}); \tag{19}$$

$$R_0 = ss' \sqrt{\frac{w}{s^2(s'^4 - 1) + s'^2(s^4 - 1)}}. \tag{20}$$

where I_2 is the region within which the summation is well-defined and M_2 is the number of (u,v) pairs in I_2. As we can see in the simulation results, the estimation based on multiple images improves the performance of our algorithm.

4. ERROR ANALYSIS

The performance of our estimation algorithm can be evaluated by introducing an additive noise in the model:

$$Y_i(u,v) = H(u,v,R_i)F_i(u,v) + N_i(u,v); \quad i = 0,1. \tag{21}$$

Combining (10), (11), (21) and the simplified version of (8) as in Section 3.1, we can give the estimation of OTF for first frame with the presence of noises as:

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

The estimation of the focused image is given by $\hat{F}_0(u,v) = Y_0(u,v)/\hat{H}(u,v,R_0)$, while the noise free estimation is $F_0(u,v) = Y_0(u,v)/H(u,v,R_0)$. This makes us arrive at the noisy version of focused image estimate as:

$$\hat{F}_0(u,v) = F_0(u,v)[\frac{Y_1(u,v) - N_1(u,v)}{Y_1(u,v)} \cdot \frac{Y_0(\frac{u}{s},\frac{v}{s})}{Y_0(\frac{u}{s},\frac{v}{s}) - N_0(\frac{u}{s},\frac{v}{s})}]^{\frac{-s^2}{s^4-1}}. \tag{22}$$

Notice that the original additive noise becomes multiplicative noise in the final estimation. The statistical characteristic of the noise also changes. The random variable inside the square bracket is the ratio of two non-zero mean normal random variables. Its distribution has been studied,[9] based on which we can study the distribution and expectation of the noise. More importantly, by making a realistic assumption that signal-to-noise ratio (SNR) of the two blur image are identical, we notice that the term inside the square bracket has value close to one. This suggests that SNR of the estimation in (22) will be high, which claims the superiority of our algorithm in terms of suppressing noise.

5. ESTIMATION OF PHASE TRANSFER FUNCTION

Section 3 presents our algorithm for estimating the OTF when it has no phase components. 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 lens, such as materials, shapes and sizes. Here we provide an approach for estimating the PTF without the knowledge of physical characteristics of the lens.

Recall from (3) and (5) that the two blurred images has the following relationship in frequency domain:

$$s^2 \frac{Y_1(u,v)}{Y_0(\frac{u}{s},\frac{v}{s})} = \frac{H(u,v,R_1)}{H(u/s,v/s,R_0)} exp\{j2\pi(u\frac{t_x}{s} + v\frac{t_y}{s})\}. \tag{23}$$

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

$$H(u,v,R_i) = |H(u,v,R_i)| exp\{j * \theta_i(u,v)\}, \quad i = 0,1. \tag{24}$$

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 Eq. (23) as the following:

$$\angle Y_1(u,v) - \angle Y_0(\frac{u}{s},\frac{v}{s}) - 2\pi(u\frac{t_x}{s} + v\frac{t_y}{s}) = \theta(u,v) - \theta(\frac{u}{s},\frac{v}{s}), \tag{25}$$

where $\angle Y_i(u,v)$ denotes the phase of the images in frequency domain. Note that the LHS of (25) is consists of known variables. Without assuming any specific forms of θ, one idea for solving the above equation is to use Taylor expansion. We apply the 2D Taylor expansion of the first order to $\theta(u,v)$ at the point $(u/s, v/s)$, and denote the LHS of (25) as $C(u,v)$ to obtain:

$$C(u,v) = (s-1)\frac{u}{s}\theta_u(\frac{u}{s},\frac{v}{s}) + (s-1)\frac{v}{s}\theta_v(\frac{u}{s},\frac{v}{s}), \tag{26}$$

where $\theta_u(\frac{u}{s}, \frac{v}{s})$ denotes the partial derivative of θ with respect to u evaluated at the point of $(\frac{u}{s}, \frac{v}{s})$. Similar interpretation applies to $\theta_v(\frac{u}{s}, \frac{v}{s})$.

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

$$D(\xi, \eta) = \xi \frac{\partial \theta(\xi, \eta)}{\partial \xi} + \eta \frac{\partial \theta(\xi, \eta)}{\partial \eta}. \tag{27}$$

A general solution[10] can be obtained in the following form:

$$\theta(\xi, \mu) = \int \frac{1}{\xi} D(\xi, \xi\mu) d\xi + \Phi(\mu), \qquad \mu = \eta/\xi. \tag{28}$$

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 $\Phi = 0$. The solution in (28) is in the form of integral, and it is approximately equivalent to summations when the variables are discrete. (In our case, we have discrete Fourier transforms). Equation (28) is thus equivalent to the following explicit equation:

$$\theta(\xi, \mu) = \sum_{i=0}^{\xi} \frac{1}{i} D(i, i\mu). \tag{29}$$

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 $\eta = v/s$.

6. SIMULATION RESULTS

We test the effectiveness of our algorithm in video sequences captured by a digital camcorder. The sequences are captured in a frame rate of 10fps and with a resolution of 320×240. In all sequences, the camcorder moves towards the objects and along the optical axis, i.e., no rotations are presented. The camcorder used here has a large focus, namely the focus depth is far from the lens. In this case, when depth decreases the blur radius will increase.

Fig.1 (a) shows the frame 1, 3, 5, 7 and 9 of the video sequence *BOOK*, in which the planar object is a book positioning parallel to the camera. Fig.1 (b) shows the sequence blurred by a simulated Geometric blur as in (14) with blur radius for frame 1 as $R_1 = 8$ (in pixel). Fig.1 (c) is the reconstruction for frame 1 using increasing number of frames, i.e., the second image is the estimated focused frame 1 computed using frame 1, frame 2 (not shown) and frame 3 in the blurred sequence (b). It is easy to notice that multiple-frame estimation performs much better than two-frame estimation. The estimated frames become very close to the original frames when the number of frame used exceeds 5. This is further demonstrated in Fig. 4 (a), which shows the blur radius estimation versus groundtruth. The black solid line represents the true blur radius for frame 1 $R_1 = 8$, with two black dashed line representing $\pm$ 5%R_1 within which the reconstruction will have 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, namely, it is an accumulative average of the blur line. As we can see, although the two-frame estimation varies from -25%R_1 to 25%R_1 among different frames, multiple-frame estimation gives steady results and lies within $\pm$ 5%R_1 after frame number exceeds 6.

Fig.2 (a) shows the frame 6, 56, 106, 156, 206 and 256 of a long video sequence *SOCCER*, in which the object soccer is an approximately planar object. Fig.2 (b) shows the sequence blurred by a simulated Gaussian blur as in (10) with a linearly increasing blur radius from $R_6 = 4$ to $R_{206} = 12$. The increasing blur radius models the effects of decreasing depth caused by forward camera motion. Fig.2 (c) is the reconstructed sequence where 5 frames are used for estimating each frame. As can be seen that the estimation of focused sequence gives constantly good performance over a large number of frames.

Fig.3 (a) shows the frame 1, 31, 61, 91 and 121 of a 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

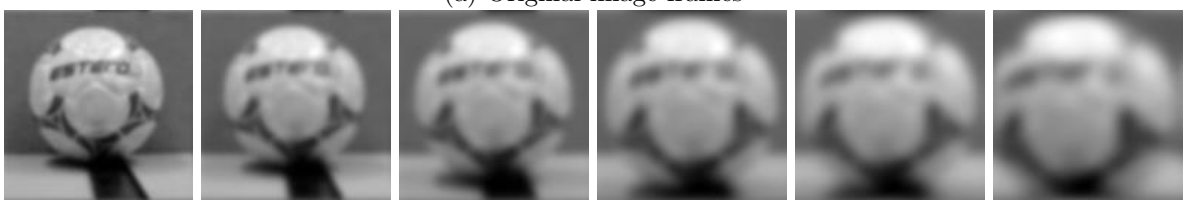
(a) Original image frames

(b) Blurred image frames

(c) Reconstructed image frames

Figure 1. Estimation results for frame 1 in video *BOOK*, in the case of a geometric blur PSF.

(a) Original image frames



(b) Blurred image frames

(c) Reconstructed image frames

Figure 2. Estimation results for frame 6, 56, 106, 156, 206 and 256 in video *SOCCER*, in the case of a Gaussian blur PSF.

in Fig.3 (b). As mentioned in Section 2, our algorithm can be applied to smaller 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

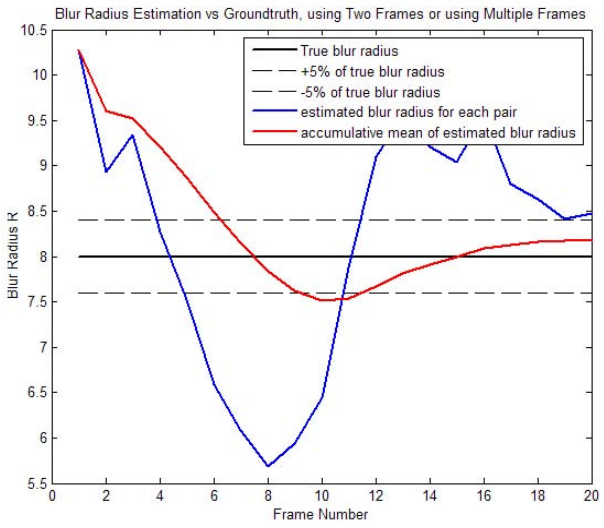
regions and foreground regions. Fig.3 (c) is the reconstructed sequence where we use 8 frames for estimating each frame. It can be seen that our algorithm gives high-quality reconstruction for both the foreground objects and the background. Fig. 4(b) is a plot with the groundtruth blur radius for background (solid black line) and the one for foreground (black line with dots). Similarly as in *SOCCER*, 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 robustness of our algorithm in dealing with 3D scenes.

(a)Original image frames

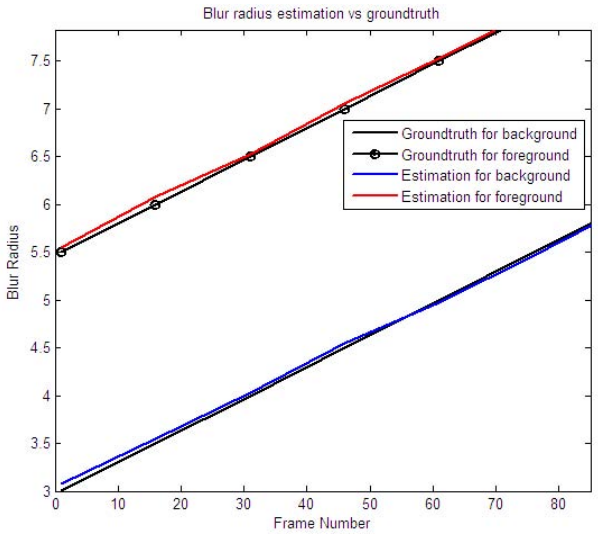
(b)Blurred image frames

(c)Reconstructed image frames

Figure 3. Estimation results for frame 1, 31, 61, 91 and 121 in video *DESK*, in the case of a Gaussian blur PSF.



(a) Estimated Blur radius for sequence *BOOK* versus groundtruth, in the case of two-frame estimation and multiple-frame estimation



(b) Estimated Blur radius for sequence *DESK* versus groundtruth, in the case of different blur radiuses for background and foreground.

Figure 4. Plots for estimated blur radiuses versus groundtruths for video sequence *BOOK* and *DESK*.

7. CONCLUSIONS

In this paper, we introduced a novel method for focused image sequence estimation. The proposed algorithm relies on the differences in blur characteristic of multiple images resulting from camera motion in video sequences. Noise analysis suggests the estimation improves SNR while computer simulation proves the competence of our approach. Our technique could potentially be used to replace the expensive apparatus required for auto-focus adjustments by miniature engines or sensors in many camera devices.

ACKNOWLEDGMENTS

The authors would like to thank the Physical Realization Research Center of Excellence, Motorola Labs, for supporting this work.

REFERENCES

1. M. Subbarao, "Efficient depth recovery through inverse optics," *Machine Vision for Inspection and Measurement (H. Freeman, Ed.), New York: Academic* , 1989.
2. J. Ens and P. Lawrence, "An inverstigation of methods for determining depth from focus," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15**, pp. 97–108, Feb. 1993.
3. J. Rayala, S. Gupta, and S. Mullick, "Estimation of depth from defocus as polynomial system identification," *IEE Proceedings* , pp. 356–362, 2001.
4. A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26**, pp. 1521–1525, Nov. 2004.
5. A. Levi and H. Stark, "Image restoration by the method of generalized projections with application to restoration from magnitude," *Journal of the Optical Society of America A* **1**, pp. 932–943, Sep. 1984.
6. J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, "Online video stabilization based on particle filters," in *IEEE International Conference on Image Processing*, (Atlanta, GA), Nov. 2006.
7. R. Bracewell, K. Chang, A. Jha, and Y. Wang, "Affine theorem for two dimensional fourier transform," *Electronics Letters* , Feb. 1993.
8. F. Bowman, *Introduction To Bessel Functions*, Dover Publication Inc., New York, 1958.
9. D. Hinckley, "On the ratio of two correlated normal random variables," *Biometrika* , 1969.
10. A.D.Polyanin, V.F.Zaitsev, and A. Moussiaux, *Handbook of first order partial differential equations*, Taylar and Francis, New York, 2002.