ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

3D Scene Recovery from Defocused Video Sequence

Mrs. Preeti N. Baheti, Prof. N. N. Kasat, Prof. S. Kasturiwala

Dept.of EXTC, SIPNA's COET, Maharashtra, India
Dept.of EXTC, SIPNA's COET. Maharashtra, India
Dept.of EXTC, SIPNA's COET. Maharashtra, India

ABSTRACT—***The paper aims at clearing the out of focus blur. Lens defocus introduces uniform blurring, Restoring these kinds of images is an interesting research field .The low resolution cameras manufactured at lower end like mobile phone and web camera mostly suffers through out-of-focus blur. Current auto focus solutions used in commercial cameras are design to ensure that captured images are in focus by adjusting the lens position and aperture opening. This needs an auto focusing hardware. The assembly is bulky and costly, cannot employed at handheld application where cameras are rigid (fixed lens &aperture). Moreover it has fundamental limitations that when the scene contains multiple objects with largely varying depths , a single image cannot capture all the objects in focus simultaneously in that case camera captures multiple images with variable depth. We present a novel method for virtually focusing the defocused video captured by camera. Virtual Focusing stands for the technique of providing an image processing solution to recover focused image sequences from videos taken by an out-of-focus camera with fixed physical parameters (cell phone cameras and webcams).***

Keywords— **Depth from Defocus, PSF Modeling, Blur radius Estimation, image restoration.**

I. Introduction

Focusing is an important issue in digital camera design. Current auto-focus solutions widely applied in industry are mostly based on different focus measures. They search for the best focused images while moving the lens and can be tuned to perform fast. The shortage is that they require a focal-length changing lens and an accurate engine that can move the lens with a particular step size. From a different point of view however, image processing solutions model the out-of-focus phenomenon as focused images passing through a linear system. With the estimation of the point spread function (PSF) of the lens system, the focused images can be recovered through a deconvolution process. Discussion in this paper will lie mainly within this class and concentrate on PSF estimation. The overall philosophy of estimating PSF and its Fourier Transform, also known as optical transfer function (OTF), is based on a fundamental observation that is the blur characteristic relates only to the object depth and the camera settings. Despite the fact that the relationship is usually approximated by first order optics, this observation verifies itself through The success of depth-from-defocus (DFD) algorithms. For example, designer in [1] utilizes two settings of camera parameters for acquiring two differently blurred images. Assuming the PSF to be a Gaussian function, a close form solution of the blur parameter is given. More generally, the authors of [2] approximate the underlying OTF by a parametric polynomial and estimate the coefficients using a least-square criteria. In a more recent work [3], the technique of DFD is combined with stereo pairs and the estimation is performed with the tool of 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 experimental device, 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 mobile-phone cameras. The other novelty of our algorithm is to exploit multiple images taken by a moving camera. Multiple images taken in variable positions not only provide differently blurred images but also reveal additional resources for improving estimation.

The rest of this paper is organized as follows. In Section 2, we begin with the problem definition and model formulation. In Section 3, we explain the main idea of blur estimation through two examples of PSF. We then discuss in Section 4 the idea of multiple-image estimation and in Section 5 noise analysis for the system. Section 6 provides the simulation results and Section 7 draws the conclusion followed with Refernces.

II. Camera and Imaging Model

Assume a moving camera is looking at a static object and taking a video of it. 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 [4]

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

$$\begin{bmatrix} x1 \\ y1 \end{bmatrix} = \frac{z0-f}{z1-f}\begin{bmatrix} r11 & r12 \\ r21 & r22 \end{bmatrix}\begin{bmatrix} x0 \\ y0 \end{bmatrix} + \begin{bmatrix} ty \\ ty \end{bmatrix} \quad (1)$$

Define $s \triangleq (z0-f)/(z1-f)$ which represents the scaling factor between the two mages 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 2D Fourier transform , $F_0(u,v)$ and $F_1(u.v)$ have the following relationship:

$$F1(u,v) = \frac{1}{s^2} F0\left(\frac{u}{s}, \frac{v}{s}\right) exp\left\{j2\pi\left(tx\frac{u}{s} + ty\frac{v}{s}\right)\right\} \quad (2)$$

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):*

$$Y_i(u,v) = F_i(u,v)H(u,v,Ri) \quad \mathrm{i} = 0,1 \quad (3)$$

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 (Equation 5):

$$s^2 Y_1(u,v) = Y_0\left(\frac{u}{s}, \frac{v}{s}\right) . \frac{H(u,v,R1)}{H(u,v,R0)} \quad (5)$$

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

$$R_i = \lambda L\left(\frac{1}{f} - \frac{1}{Z_i} - \frac{1}{\lambda}\right) \qquad i = 0,1; \quad (6)$$

which is equivalent to

$$Z_i = \left(\frac{1}{f} - \frac{1}{\lambda}\left(1 + \frac{R_i}{L}\right)\right)^{-1} \qquad i = 0,1\,; \quad (7)$$

where λ 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 object depth once the camera parameters are fixed. From the definition of *s* we can continue to write as a function of the blur radiuses R_0 and R_1

$$s = \frac{Z_0 - f}{Z_1 - f} = \frac{R_0 + L}{R_1 + L} \times \frac{R_1 + L - \lambda L/f}{R_0 + L - \lambda L/f} \quad (8)$$

To recover the focused images from the blurred images, we need to estimate the OTF, which equals identifying the blur radiuses. With A, *L*, *f* being known camera parameters, we will see in the following section that based on (Equation 8) and (Equation 9), it is able to solve for s , R_0 , R_1 thus for $H(u,v,R_0)$ and $H(u,v,R_1)$.

III. BLUR ESTIMATION

In this section, we will discuss our algorithm for three types of PSF. In all the cases, we begin with assuming the energy conservation constraint, which means $H(0,0,R) = 1$. Thus, *s* can be solved by noticing the DC components in (5) yields

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

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

$$Z(u,v) \triangleq s^2 \frac{|Y_1(u,v)|}{\left|Y_0\left(\frac{u}{s}, \frac{v}{s}\right)\right|} \quad (10)$$

Therefore, using (9), we obtain

$$Z(u,v) = \frac{H(u,v,R_0)}{H(u,v,R_1)}$$

$Z(u,v)$ 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.

Gaussian Blur Model

Two frames algorithm for blur estimation

Step1—Write function FrameExtract() to extract frames from blur video
Step 2-- Take the FFT of 1st two blurred video frames I and I1 and name them as Y0 and Y1
Step 3--Calculate Scaling factor S

$$s = \sqrt{\frac{Y0(0,0)}{Y1(0,0)}}$$

- Where Y0(0,0) and Y1(0,0) are the dc component ie amplitude at Zero frequency
- S will be purely scalar quantity

Step 4--Compute intermediate term Z(u,v), where Z(u,v) is the ratio of two corresponding frequency components of two consecutive frames selected for estimation

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

Step 5--Compute constant C which is further used to estimate R_0 equal to

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

$$c = \frac{1}{M_1}\sum_{(u,v)\varepsilon I_1} \frac{-4}{u^2+v^2} s^2 \frac{|Y_1(u,v)|}{\left|Y0_0\left(\frac{u}{s},\frac{v}{s}\right)\right|}$$

$$R_1^2 - R_0^2/s^2 = c$$

Where I1 is the region where the summation is well-defined, which mainly excludes frequencies where the absolute value of frequency component has zeros values or values close to zero.M1 is the number of(u,v) pairs in I1.

Step 6-- For Gaussian model R_1=**s**R_0 Once we compute C we can estimate R0

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

Step 7-- Once we have blurring radiuses R0 & R1 we can createOTF for Gaussian using

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

Step 8-- Convert OTF to PSF using **otf2psf()** function in Matlab since we were working in frequency domain and deconvolution algorithm works only on PSF.

A. *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. Therefore we have

$$H(u,v,R) = 2\frac{J_1\left(R\sqrt{u^2+v^2}\right)}{\left(R\sqrt{u^2+v^2}\right)} \quad (12)$$

TWO FRAMES ALGORITHM FOR BLUR ESTIMATION

Step 1-- Extract frames from blur video

Step 2-- Take the DFT of 1st two blurred video frames name them as Y0 and Y1

Step 3-- Calculate Scaling factor s

$$s = \sqrt{\frac{Y0(0,0)}{Y1(0,0)}}$$

- Where Y0(0,0) and Y1(0,0) are the dc component ie amplitude at Zero frequency
- S will be purely scalar quantity

Step 4--Compute Z(u,v)

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

Step 5-- $\boldsymbol{Z(u,v)} \triangleq 1 + a_1(u^2+v^2) + a_2(u^2+v^2)^2 + \ldots\ldots.,$
Solve the matrix Z=Ua to find out least mean square value for a=$[a_1, a_2 \ldots\ldots. a_N]$

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

Z=Ua $\qquad a = [U^T U^0]^{-1} U^T z$

step 6 – once we have array of 'a' determined we can easily find out R0 and R1

$a_1 = -\frac{1}{8}(R_1^2 - R_0^2/s^2);$ $\qquad a_2 = -\frac{1}{192}(R_1^4 - R_0^4/s^4) + \frac{R_0^2}{8s^2}a_1;$ ...

N=3 gives enough accuracy.

Step 7—Create OTF using R0 and R1 substituted in following equation.

$$H(u,v,R) = 2\frac{J_1\left(R\sqrt{u^2+v^2}\right)}{\left(R\sqrt{u^2+v^2}\right)}$$

Step 8-- Convert OTF to PSF using **otf2psf()** function in Matlab since we were working in frequency domain and deconvolution algorithm works only on PSF .

IV. VIDEO RECONSTRUCTION

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 (Equation 11). In the case of 3D 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.

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.

MULTIPLE FRAME ALGORITHM FOR ESTIMATION OF BLUR RADIUS

Step 1-- Extract frames from the video.consider first L frames

Step 2--Take FFT of first and i[th] frame where i=1,2,3,…… L-1.i.e. Y0(u,v) and Yi(u,v)

Step 3-- Calculate Scaling factor S using

$$\frac{R_i}{R_0} = s_i = \sqrt{Y_0(0,0)/Y_i(0,0)}$$

Step 4-- Compute Z(u,v), where Z(u,v) is the ratio of two corresponding frequency
components of two consecutive frames selected for estimation

$$Zi(u,v) = si^2\ \frac{Yi(u,v)}{Y0(u/s, v/s)}$$

Step 5--Calculate the intermediate term W(u,v)which is further utilize to calculate the blur radius

$$W(u,v) \triangleq \frac{1}{L-1}\sum_{i=1}^{L-1}\frac{S_i^2}{S_i^2-1}\ln[Z_i(u,v)]$$

Step 6--W(u,v) is then used in the equation below to find R_0

$$R_0^{(0:L-1)} = \sqrt{\left(\frac{1}{M2}\sum_{(u,v)\varepsilon I_2}\frac{-4}{u^2+v^2}W(u,v)\right)}$$

Step7--Once we have $R_0^{(0:L-1)}$create Otf using

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

Step 8-- Convert OTF to PSF using **otf2psf()** function since deconvolution algorithm works in time domain only.

---

where I_2 are defined as regions where the absolute value of the frequency spectrum at frequency component . *M2* is the number of (u, *v)* pairs in I_2. $R_0^{(0:L-1)}$denotes the estimate using 0 to *L*- 1 frame. 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

$$\left(R_0^{(0:L)}\right)^2 = \frac{L-1}{L}\left(R_0^{(0:L-1)}\right)^2 + \frac{1}{L}\left(R_0^{(L)}\right)^2 \qquad (16)$$

where $R_0^{(0:L-1)}$ 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.

A. *Video Reconstruction*

In the light of multi-frame blur estimation idea, it becomes natural to exploit the effects of additional frames in image reconstruction. We further find that the multi-frame 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 (to ensure casualty) *N* frames. Recall that (Equation 4.8) and (Equation 4.9) can be rewritten for *N* frames as

$$F_i(u,v) = \frac{1}{s_i^2}F_0\left(\frac{u}{s_i},\frac{v}{s_i}\right)exp\left\{j2\pi\left(t_{xi}\frac{u}{s}+t_{yi}\frac{v}{s}\right)\right\}$$

$$i = 1, \dots \dots N;$$

And $\quad Y_i(u,v) = F_i(u,v)H(u,v,Ri), \qquad i = 0,1, \dots N$

Together they yield

$$Y_i(us_i, vs_i) = \frac{1}{s_i^2}exp\{j2\pi(t_{xi}u + t_{yi}v)\}$$

$$H(us_i, vs_i, Ri)F_0(u,v)$$

$$i = 1, \dots \dots N; \qquad (17)$$

Since and *H(.)* have been obtained previously, we denote

$$Y_i^{'}(u,v) \triangleq Y_i(us_i, vs_i)\ \&$$

$$G_i(u,v) \triangleq \frac{1}{s_i^2}exp\{j2\pi(t_{xi}u + t_{yi}v)\}H(us_i, vs_i, Ri)$$

Thus equation becomes

$$Y_i^{'}(u,v) = G_i(u,v)F_0(u,v) \qquad i = 1, \dots \dots N; \qquad (18)$$

which is a standard 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,

$$\widehat{F_0}(u,v) = \frac{\sum_{i=1}^{N} G_i^*(u,v)Y_i^{'}(u,v)}{\sum_{i=1}^{N}\left|G_i^*(u,v)\right|}$$

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 which is however beyond the scope of our work.

VI. ERROR ANALYSIS

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

$$Y_i(u,v) = F_i(u,v)H(u,v,Ri) + N_i(u,v) \qquad i = 0,1 \qquad (19)$$

Using the Gaussian blur model and simplified version of (7) as in Section 3.1, we can give the estimation of OTF for first image with presence of noise as:

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

$$\widehat{H}(u,v,R_0) = \left[s^2 \frac{Y_1(u,v)-N_1(u,v)}{Y_0(u/s,v/s)-N_0(u/s,v/s)}\right]^{\frac{s^2}{s^4-1}} \qquad (20)$$

The estimation of the focused image is given by ˆ $F0(u,v)$ = $Y0(u, v)/\hat{H}(u\ v,R0)$, while the noise free estimation is $F0(u,v)$ = $Y0(u,v)/H(u,v,R0)$. This makes us arrive at the noisy version of focused image estimate as:

$$\widehat{F_0}(u,v) = F_0(u,v)\left[\frac{Y_1(u,v)-N_1(u,v)}{Y_0(u/s,v/s)-N_0(u/s,v/s)} \cdot \frac{Y_0(u/s,v/s)}{Y_1(u,v)}\right]^{\frac{-s^2}{s^4-1}}$$

Notice that the original additive noise becomes multiplicative noise in the final estimation. The statistical characteristic of the noise also changes. If we assume the noises $N_0(u,v)$ and $N_1(u,v)$ have Gaussian distributions, thenThe random variable inside the square bracket is the ratio of two normal random variables with non-zero mean . Based on that, we can also give 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 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.

VII. SIMULATION RESULT

We test the effectiveness of our algorithm with synthetically blurred video sequences as well as real blur sequences. For synthetic blur tests, video sequences are captured by a web camera. The original sequences are captured in a frame rate of 15 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. Moreover, in all sequences, the webcam moves towards the objects (unless otherwise notified) within a depth range between 500 and 2000 mm . The camera moves with a relatively low speed of approximately 30 mm per second so that motion blur has been introduced.. Synthetic blur radii (in pixel unit) are computed according to in (7) with$f = 22$ mm, λ=22.2 mm and L=240 pixels.

A. *Color video frames , Synthetic Geometric Blur*

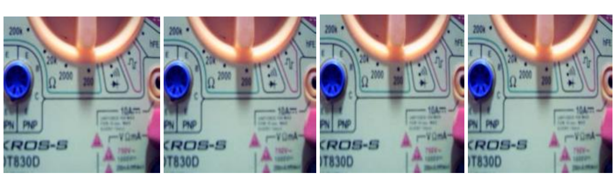

Fig.1(a)

Fig.1(b)

Fig shows comparison between the video ***MultiMeter*** blurred by a synthetic geometric blur model 1(a) and the reconstructed video frame 1(b) using 5 frames for reconstruction of video frames.

Fig. 1(a) shows the sequence blurred by a simulated Geometric blur as in with blur radius is in pixel. try to estimate the blur radii and reconstruct focused image. Fig. 1(b) shows a series of reconstructed image for 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. As we can see, although the two-frame estimations varies from different frames used, multiple frame estimation gives steady results within after frame number exceeds 6.

B. *Color video framess, Synthetic Gaussian Blur*

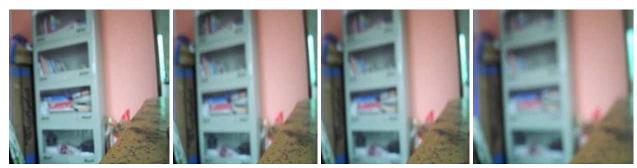

Fig.2(a)

Fig.2(b)

Fig shows comparison between video of ***Room*** blurred by a synthetic gaussian blur model (a) and the reconstructed (b) using 5 frames for reconstruction of video frames.

This video sequence consists of color images. 3(a) shows the sequence blurred by a simulated Gaussian blur. 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. Our algorithm can be applied to local regions of the image to ensure each region has the same depth.

E. Real Blurred Sequence

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

Fig 3(a)

Fig 3(b)

Fig 3(c)

Fig shows comparison between real blur video of ***Cordless*** 3(a) , 3(b) shows reconstructed video using Geometric reconstruction model using 10 frames for reconstruction and 3(c) Reconstructed video using Gaussian reconstruction model using 10 frames for reconstruction

We also test our algorithm with real blur image sequences. The sequences are captured by a web camera whose lens can be manually readjusted, 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 the camera. The physical parameters and are not available after adjustments; thus, the simplified camera geometry will be used.

The sequences are captured in a frame rate of 10 fps and with a resolution of 352× 288. Fig. 3(a) shows the Frames video sequenc*e*, in which the object is a picture of cordless phone 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. 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. Fig.3(b)&(c) shows corresponding reconstructed frames using the proposed multiframe blur estimation and multiframe image reconstruction. Five immediately preceding frames are used for reconstructing each frame. Since the form of the blur function is unknown, we provide reconstruction results based on the two different models When implemented in MATLAB on a PC with a single 1.86 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

The Proposed algorithm is a novel method for virtual focus and object depth estimation from defocused video. 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 estimation improves the SNR. 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 needed 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.

IX. REFERENCES

[1] M. Subbarao, ”Efficient depth recovery through inverse optics”, *Machine Vision for Inspection and Measurement (H. Freeman, Ed.), New York: Academic*, 101-126, 1989.

[2] J. Rayala, S. Gupta, and S. Mullick, ”Estimation of depth from defocus as polynomial system identification”, *IEEE Proceedings*, pp.356-362, 2001.

[3] A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, ”Depth estimation and image restoration using defocused stereo pairs”, *IEEE Trans. on PAMI*, vol.26, no.11, pp.1521-1525, Nov. 2004.

[4] R. Bracewell, K. Chang, A. Jha, and Y. Wang, ”Affine theorem for two dimensional fourier transform”, *Electronics Letters*, Feb. 1993.

[5] Levin, A., Fergus, R., Durand, F., and Freeman, W.: Image and depth from a conventional camera with a coded aperture. In Proceedings ACM SIGGRAPH, 2007.

[6] Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., and Tumblin, J.: Dappled photography Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Trans, on Graphics, 26(3):69, 2007.

[7] Zhou, C. and Nayar, S. K.: What are good apertures for defocus deblurring? In IEEE International Conference on Computational Photography, Apr. 2009.

[8]Zhou, C., Lin, S., and Nayar, S. K.: Coded aperture pairs for depth from defocus. In IEEE International Conference on Computer Vision, Oct. 2009.

[9] Raskar, R., Agrawal, A., and Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans, on Graph, 25(3):795-804, 2006.

[10] Farsiu, S., Robinson, M. D., Elad, M., and Milanfar, P.: Fast and robust multiframe super resolution. IEEE Trans, on Image Processing, 13(10):1327-1344, 2004.

[11] Nagahara, H., Kuthirummal, S., Zhou, C., and Nayar, S. K.: Flexible depth of field photography. In European Conference on Computer Vision, 2008.

[12] Robertson, M., Borman, S., and Stevenson, R.: Estimation-theoretic approach to dynamic range enhancement using multiple exposures. Journal of Electronic Imaging, 12(2), Apr. 2003.

[13] Yuan, L., Sun, J., Quan, L., and Shum, H. Y.: Image deblurring with blurred/noisy image pairs. ACM Trans, on Graphics, 26(3):1, 2007.

[14] Farsiu, S., Elad, M., and Milanfar, P.: Video-to-video dynamic super-resolution for grayscale and color sequences. EURASIP Journal on Applied Signal Processing, 2006:1-15, 2006.

Preeti N. Baheti received the B.E. degree in Electronics and Telecomm. in 2006 from Pune University, Maharashtra, India. She is currently pursuing the M.E. degree in Digital Electronics. University of Amaravati, India. She worked as software Engg. In Synergy Infoser's from 2006 to 2007. She has been lecturer in HVPM's COET from 2008 to 2009.

Prof. N. N. Kasat received his B.E. in E&TC degree in 1998 from Govt. COE, Amaravati. Post Graduated from Badnera COE. In 2008. Currently pursuing PhD in the area of Embedded Systems & Parallel Computing. He is working as a Professor in SIPNA COET,Amravati. He is the founder and Incharge of IETE Student Forum (ISF) at Sipna's COE. Elected as Secretary for IETE Amaravati Centre. In 2010.

Prof. Sanket B. Kasturiwala, He graduated in E&TC from the Sipna COET, Amravati University , INDIA in 2005. And postgraduated in Digital Electronics from S.S.G.M.C.E., Shegaon from Amravati University , Amravati in 2007.

He is Assistant Professor in department of. E&TC from the Sipna COET, Amravati – Maharashtra (INDIA) since last three years, where he is pursuing his Ph.D. degree in the area of image processing under the supervision of Dr. S.A.Ladhake, Principal, Sipna College of Engg. & Tech., Amravati – Maharashtra (INDIA). He is also the member of professional bodies such as IETE, ISTE, IAENG.