Real-time restoration algorithm based on one-dimensional Wiener filters for different rates of image motion blur

Shi Li
Chinese Academy of Sciences
Changchun Institute of Optics
Fine Mechanics and Physics
Changchun 130033, China
and
Graduate School of the Chinese Academy of Sciences
Beijing 100039, China
E-mail: brightlishi@gmail.com

Bao Zhang
Hui Sun
Chinese Academy of Sciences
Changchun Institute of Optics
Fine Mechanics and Physics
Changchun 130033, China

Abstract. *To eliminate side-oblique image motion, a fast image algorithm is proposed for implementation on aerial camera systems. When an aerial camera works at a side-oblique angle, much parallel image motion with different rates will occur on the focal plane array simultaneously. Through analysis of how different rates of parallel image motion blur are generated and the relationship between image motion and the field of view (FOV) angle in side-oblique situations, the entire blurred image can be segmented into many slices by their different rates of image motion. To be computed quickly, the slices are divided into pixel lines continuously, and then a specific parallel computing scheme is presented using 1-D Wiener filters to restore all the pixel lines. With all the resulting pixel lines combined, the restoration image comes into being. The experiment results show that the proposed algorithm can effectively restore the details of side-oblique blurred images. The peak signal-to-noise ratio (PSNR) of the restored image can reach 31.426. With the help of the parallel computing capability of a graphics processing unit (GPU), the proposed algorithm can restore a* 2048×2048 8-bit *blurred image in* 17 ms*, realizing real-time restoration. © 2009 SPIE and IS&T.* [DOI: 10.1117/1.3125982]

1 Introduction

Image motion is inevitable when an aerial camera works on high-velocity, low-altitude (highV/H) airborne platforms. The complicated and changing flying attitudes of aircraft will influence the working attitude of aerial cameras, which can create various kinds of image motion. This type of image motion is associated with the specific attitude of the aerial camera and the roll, pitch, and yaw rates of aircraft during the exposure time. When an aerial camera works at a side-oblique angle on a forward flight aircraft, the image motion in different regions of the focal plane array is different, forming different rates of image motion blurred images.

There are many approaches to compensate for side-oblique motion blur, such as on-chip electronic forward motion compensation (FMC),[1] moving focal plane,[2,3] controlling focal plane shutter,[4] and so on. Unfortunately, charge-coupled device (CCD)/complementary metal oxide semiconductor (CMOS)-incorporated on-chip electronic FMC is too expensive to be widely applied. Also, the approaches of motion compensation by moving focal plane or controlling focal plane shutter get very bulky and have complicated mechanism structures that consume much payload and container capacity of the aircraft.

Taking cost and bulk weight into consideration, it can a good choice to utilize digital image approaches to eliminate side-oblique motion blur. But most restoration algorithms are designed to compensate the identical rate of image motion.[5,6] In this work, a novel algorithm is proposed where the entire blurred image is segmented into many slices by their different rates of image motion. Each slice in the frames has an identical rate of image motion and is matched with a unique point spread function (PSF). The deductive process of the algorithm is described in Sec. 2, and the experiment results are shown in Sec. 3.

2 Description of the Algorithm

2.1 *Analysis of Side-Oblique Image Motion*

Figure 1 illustrates the different rates of image motion produced when an aerial camera works at a side-oblique angle.

Paper 08081R received May 25, 2008; revised manuscript received Feb. 9, 2009; accepted for publication Mar. 23, 2009; published online Apr. 29, 2009.

1017-9909/2009/18(2)/023005/6/$25.00 © 2009 SPIE and IS&T.

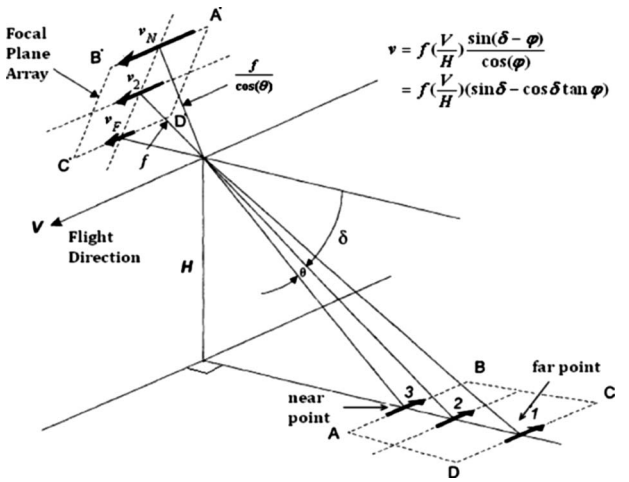


Fig. 1 Side-oblique image motion compensation analysis.

v_N and v_F represent the image motion speed of the near and far points on the focal plane array's surface, respectively. From the simple geometrical relationship in Fig. 1, v_N and v_F can be calculated as follows:

$$v_F = f\left(\frac{V}{H}\right)\frac{\sin(\delta-\theta)}{\cos(\theta)}, \tag{1}$$

$$v_N = f\left(\frac{V}{H}\right)\frac{\sin(\delta+\theta)}{\cos(\theta)}, \tag{2}$$

where V is the aircraft velocity, H is the aircraft altitude, f is the camera's focal length, δ is the camera's depression angle, and θ is the camera's FOV half angle.

The speed ratio of the image motion between near point and far points is

$$\frac{v_N}{v_F} = \frac{\sin(\delta+\theta)}{\sin(\delta-\theta)}.$$

When θ is a fixed value, the ratio v_N/v_F will increase as the δ decreases in the zone of $[90\ \text{deg}, \theta)$ and the range of the ratio is $[1, +\infty)$. It is implicated that the difference between v_N and v_F cannot be neglected, owing to the existence of depression angle δ.

To describe the image motion rate of general points conveniently, $\varphi(-\theta \le \varphi \le \theta)$, the FOV angle of the current corresponding object point, is defined. Moreover, the counterclockwise direction is defined as the positive direction. The image motion rate of a general point is described as follows:

$$v = f\left(\frac{V}{H}\right)\frac{\sin(\delta-\varphi)}{\cos(\varphi)}. \tag{3}$$

If exposure time T and the pixel size of the CCD/CMOS c are given, the displacement of general blurred points associated with the view angle φ can be calculated as follows:

$$\tilde{L} = f\cdot\frac{1}{c}\cdot\left(\frac{V}{H}\right)\cdot T\cdot\frac{\sin(\delta-\varphi)}{\cos(\varphi)} = f\cdot\frac{1}{c}\cdot\left(\frac{V}{H}\right)\cdot T(\sin\delta - \cos\delta\tan\varphi). \tag{4}$$

From Eq. (4) it can be seen that when δ is determined, $\tilde{L}$ will decrease as φ increases in the range of $[-\theta, \theta]$. One $\tilde{L}$ may produce one PSF, and many different $\tilde{L}s$ will produce many different PSFs. To restore the single blurred image with many different PSFs, we segment the entire blurred image into many slices to match each PSF.

2.2 *Image Segmentation*

2.2.1 *Point spread function configuration*

From the monotone decreasing relationship between displacement $\tilde{L}$ and view angle φ in Eq. (4); the image displacement of the near point in the side-oblique aerial images is known as the maximal one, and the far point displacement is the minimum. The expression is as follows:

$$\tilde{L}_{\min} = L_F = f\cdot\frac{1}{c}\cdot\left(\frac{V}{H}\right)\cdot T\cdot(\sin\delta - \cos\delta\tan\theta), \tag{5}$$

$$\tilde{L}_{\max} = L_N = f\cdot\frac{1}{c}\cdot\left(\frac{V}{H}\right)\cdot T\cdot(\sin\delta + \cos\delta\tan\theta). \tag{6}$$

For digital image processing, $\tilde{L}$, $\tilde{L}_{\min}$, and $\tilde{L}_{\max}$ in Eqs. (4)–(6) are rounded to the nearest integer value and marked as L, $L_{\min}$, and $L_{\max}$, respectively. The difference ΔL is defined as follows:

$$\Delta L = L_{\max} - L_{\min}. \tag{7}$$

From GPS/INS systems in aircraft, we can obtain the parameters of aircraft velocity V, altitude H, and depression angle δ, live. Also, with camera focal length f and camera FOV half angle θ, L, $L_{\min}$, $L_{\max}$, and ΔL can be calculated immediately. Hence, we can get $\Delta L+1$ different displacements in each frame dynamically:

$$L_n = L_{\min} + n \quad (0 \le n \le \Delta L). \tag{8}$$

From Eq. (8); the $\Delta L+1$ PSFs in the same frame are obtained as:

$$h(x,n) = \begin{cases} \dfrac{1}{L_n}, & x = 0, 1\cdots L_n - 1 \\ 0, & \text{elsewhere} \end{cases}. \tag{9}$$

n in $h(x,n)$ shows that the current PSF is number $n+1$ PSF in the frame. The range of n is $[0, \Delta L]$.

2.2.2 *Image segmentation and point spread function distribution*

As described in the last section, $\tilde{L}$ in Eq. (4) is discretized to L, and all $\tilde{L}s$ values among $[L-1/2, L+1/2)$ are rounded to L, forming a region with same displacement in the blurred image. In accordance with Eq. (4), the view angle φ_1 and φ_2 can be calculated to match the corresponding displacement $L-1/2$ and $L+1/2$, respectively. Between

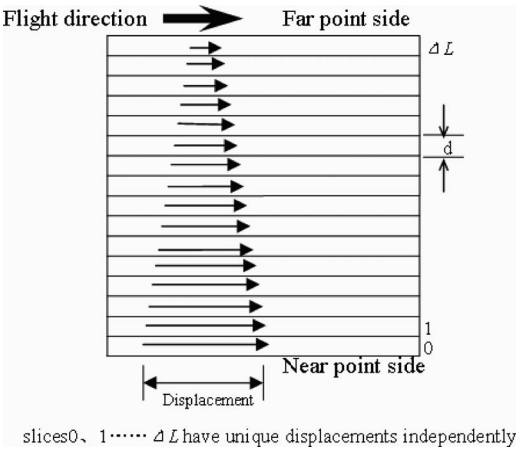


Fig. 2 Segmentation of side-oblique image.

view angles φ_1 and φ_2, the image point will form a rectangle image zone that has an identical image displacement. In light of that, we segment the side-oblique aerial image into $\Delta L+1$ parallel image slices (as in Fig. 2), with each image slice having a unique PSF function based on Eq. (9).

From Fig. 3, the width of each slice can be calculated as follows:

$$d = \frac{f}{c} \cdot (\tan \varphi_1 - \tan \varphi_2), \tag{10}$$

where d is the count of pixels contained in the width length. From Eq. (4), it can be deduced that:

$$\left(L + \frac{1}{2}\right) - \left(L - \frac{1}{2}\right) = f \cdot \frac{1}{c} \cdot \left(\frac{V}{H}\right) \cdot T \cos \delta (\tan \varphi_1 - \tan \varphi_2) = 1. \tag{11}$$

Finally, we obtain Eq. (12) from both Eqs. (10) and (11),

$$d = \frac{1}{\left(\frac{V}{H}\right) \cdot T \cos \delta}. \tag{12}$$

It can be seen from Eq. (12) that the width of slices is determined by aircraft speed V, altitude H, depression angle δ, and exposure time T. The four parameters in one aerial frame are treated as constant values. Therefore the width of each slice is equal in the same image.

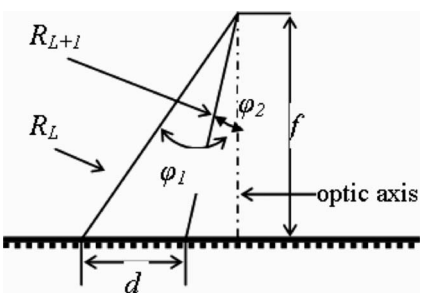


Fig. 3 Segmentation width analysis.

Fig. 4 Parallel computing.

When segmentation and the PSF of each slice are obtained, each image slice can be restored independently by a Wiener filter. However, the width of the slices in different frames may be different, leading to a dynamical configuration for 2-D fast-Fourier transform (FFT), which is the base of the Wiener filter. If the transform size of the FFT is uncertain, the FFT transform will be very inefficient and the efficiency of restoration must be limited.

2.3 *Fast Processing*

2.3.1 *One-dimensional Wiener filter*

To avoid dynamical configuration of the 2-D FFT, we use a 1-D Wiener filter, which is based on 1-D FFT, to replace the 2-D Wiener filter (WF) to restore the blurred image.[7] Here, we use 1-D WF to distinguish from the traditional 2-D Wiener filter.[5,6]

Each slice in the blurred image will be decomposed into pixel lines continuously, and each pixel line in the same slice has the same PSF function. Therefore, we can utilize a 1-D WF to restore each pixel line based on its own PSF:

$$F(u) = \frac{H(u)^*}{H(u)H(u)^* + k} G(u), \tag{13}$$

where $F(u)$, $H(u)$, and $G(u)$ denote the FFT of the restored pixel line $f(x)$, the PSF $h(x)$, and the blurred pixel line $g(x)$, respectively. The asterisk in Eq. (13) represents a complex conjugate, and k in Eq. (13) is the reciprocal of the signal-to-noise ratio (SNR) of the image.

When 1-D WF restoration is finished, all the resulting pixel lines will be combined to form the final restored image. It can be easily proved that the restoration algorithm based on 1-D FFT is 2× faster than the algorithm based on 2-D FFT.[7,8]

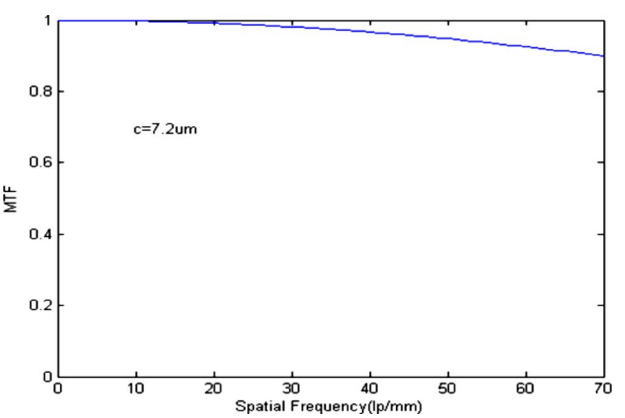
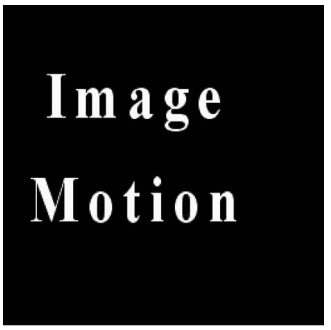
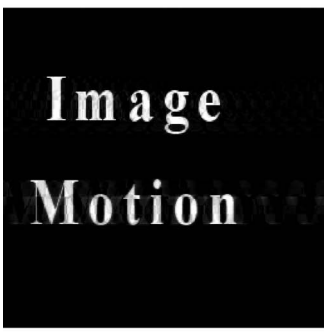


Fig. 5 MTF performance.



(a)

(b)



(c)

Fig. 6 Side-oblique blurred image synthesis. (a) The original image. (b) The blurred image with different rates of image motion. (c) The image restored by the proposed algorithm.

Fig. 7 Original image (2×2k).

Fig. 8 Sample of Fig. 7.

2.3.2 *Parallel computing*

In the proposed algorithm, there is no correlation between each of the pixel lines. Hence, each pixel line can be processed independently. Figure 4 is the parallel computing design for the algorithm. $g(n)$, $G(n)$, $F(n)$, and $f(n)$ in Fig. 4 denote the blurred pixel line, transformed blurred pixel line, transformed restored pixel line, and restored pixel line of the n'th pixel line, respectively. 1-D WF(n) represents the 1-D Wiener filter to the corresponding data in the n'th pixel line.

2.4 *Algorithm Accuracy*

In Sec. 2.2, all $\tilde{L}$ among $[L-1/2, L+1/2)$ are rounded to L for discretization. Obviously, the discretization error is introduced, and the restoration accuracy is limited by the modulation transfer function (MTF) associated with the discretization error. It is easy to calculate the difference between the displacement at the very edge of any slice and the one at the center of any slice. And the maximal discretization error δL is sure to be 0.5 pixels. The MTF due to the discretization error is defined by:

$$\mathrm{MTF} = \frac{\sin(\pi \cdot \delta L \cdot c \cdot f)}{\pi \cdot \delta L \cdot c \cdot f}, \tag{14}$$

where c is the pixel size of the CCD/CMOS (μm), and f is the spatial frequency (lp/mm). The MTF in our algorithm will not be affected by any other parameters, which are not

Fig. 9 Sample of the blurred image.

Fig. 10 Sample of the restored image.

Fig. 11 The real side-oblique motion blurred image.

shown in Eq. (14). The MTF performance of the proposed algorithm is shown in Fig. 5.

3 Experiments and Results

First, we use a blur image synthesis, shown in Fig. 6, to illustrate the different rates of image motion in side-oblique aerial images. The synthesized image in Fig. 6(b) shows that the upper part of the blurred image is more distinguishable than the lower part of the blurred image. It means that the displacement of image motion on the upper part of the blurred image is smaller than the one on the lower part of the blurred image. Namely, the higher the position is, the smaller displacement it takes place, and vice versa. This kind of blur is hard to restore by most general algorithms. From Fig. 6(c), the proposed algorithm shows good restoration for this blur.

In the next step of the experiment, the parameters of the digital camera are as follows: camera focal length 180 mm; camera FOV half angle 3.31 deg; CCD pixel size 7.2 $\times$7.2 μm; and CCD pixel count 2048$\times$2048.

The parameters of the experimental environment are defined by: the ratio of velocity and altitude (V/H) =0.3 rad/s; camera exposure time=5 ms; and camera depression angle=30 deg.

From Eqs. (5) and (6), we can obtain $\tilde{L}_{\min}=16.87$ and $\tilde{L}_{\max}=20.63$, then discretize them to $L_{\min}=17$ and $L_{\max}=21$, and get $\Delta L=4$ according to Eq. (7). Based on Eqs. (8) and (9), we obtain five image displacements (such as 17,18,…,21) and five corresponding PSFs.

Figure 7 is the 2048$\times$2048 original image used in the simulation experiment. In the aerial side-oblique image motion simulation, Fig. 7 is simultaneously blurred by the five PSFs mentioned before. From Eq. (12), it is known that the width of each slice is equal. Hence, we segment the blurred image into five slices equally to match the five unique PSFs, and utilize the parallel computing scheme, as Fig. 4 described, to restore the blurred image.

Figures 8–10 are a group of image samples from the original image, the blurred image, and the restored image of Fig. 7, respectively. These three sample images are from the same position on their own original images. The position is marked by a rectangle in Fig. 7. Compared to Fig. 8, the airplanes and other details in Fig. 9 cannot be recognized any more. Figure 10 is the restored image where most details are restored and the image sharpness is greatly enhanced. Visual inspection reveals that no noticeable interfering noise is introduced and the frame is clear. To provide an objective quality estimation, the mean square error (MSE) and peak signal-to-noise ratio (PSNR), before and after restoration, is listed in Table 1.

In the third part of the experiment, the proposed algorithm is applied on the real side-oblique motion blurred image (Fig. 11). The restored image is shown in Fig. 12. The details in Fig. 11 are well restored in Fig. 12. From Fig. 12 it can be seen that the restoration result of the real blurred image is not worse than the simulation result. It indicates that the proposed algorithm can work on the practical application.

In the experiment, the general purpose graphics processing unit (GPGPU)[9] technology is applied, and the algorithm is parallel operated on a GPU as Fig. 4 described. The GPU we used is an nVidia GeForce8800GTS (512 M). We use 2048 parallel threads in the program. The restoration time for the blurred images of different sizes is listed in Table 2. The results show good performance for real-time processing.

Table 1 Image estimation.

Image	Size	MSE	PSNR
Blurred image	2048$\times$2048	68.438	19.771
Restored image		46.825	31.426

Fig. 12 Restored image of Fig. 11.

Table 2 Restoration time for different image size.

Image size	512×512	1024×1024	2048×2048
Time(ms)	0.82	2.76	16.99

4 Discussion and Conclusions

In the algorithm, restricting the blur directions to be parallel to the rows of the image is not necessary. The blur directions can be arbitrary. As long as the blur pixel lines are obtained, the proposed algorithm can work. But according to the way in which the current devices access data, reading image data in a series in a row direction is more efficient. When the motion blur is precisely aligned with the image rows, motion blur pixels can be read in a row. It is easy to align the motion blur with the image rows by controlling an aerial camera moving precisely. Hence, row direction processing is advocated in practical restoration applications for restoration speed improvement.

In Sec. 2.2, there is a discretization to the displacements in each slice, and the discretization error is inevitable. However, it can be seen that the difference is no more than 0.5 pixels between the displacement at the very edge of any slice and the one at the center of any slice, so the restoration error is less than one pixel. This is totally acceptable. Our algorithm is based on 1-D FFT. By processing the same number of data, 1-D FFT can halve the computation requirements compared to 2-D FFT. Consequently, our algorithm is 2× faster than the scheme with 2-D FFT. The experiment results show that the proposed algorithm can realize real-time processing based on the GPU platform, and it can restore side-oblique image motion blur efficiently.

Acknowledgment

This work was supported by the Advanced Research Foundation for National Defence of China.

References

1. A. G. Lareau, "Advancements in E-O framing," *Proc. SPIE* **3431**, 96–107 (1998).
2. L. P. Zhai, M. Liu, and J. H. Xie, "Calculation of image motion velocity considering airplane gesture angle in oblique aerial camera," *Opt. Precision Eng.* **14**(3), 490–494 (2006).
3. Y. S. Xu, Y. L. Ding, H. Y. Tian, and B. Dong, "Calculation and compensation for image motion of aerial remote sensor in oblique situation," *Opt. Precision Eng.* **15**(11), 1779–1783 (2007).
4. B. A. Gorin, "Side oblique real-time orthophotography with the 9K ×9K digital framing camera," *Proc. SPIE* **5109**, 86–97 (2003).
5. M. R. Banham and A. K. Katsaggelos, " Digital image restoration," *IEEE Signal Process. Mag.* **14**(2), 24–41 (1997).
6. P. Jia, H. Sun, and B. Zhang, "Restoration of motion-blurred aerial image," *Opt. Precision Eng.* **14**(4), 697–703 (2006).
7. S. Li, H. Sun, and B. Zhang, "A method for real-time restoration of motion-blurred images," *Opt. Precision Eng.* **15**(5), 767–772 (2007).
8. S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, CA Technical Publishing (1997).
9. M. Pharr and R. Fernando, *GPU Gems2*, Addison-Wesley Professional, New York (2005).

Shi Li received his BS degree in optical and information engineering from Zhejiang University China, in 2003, and his MS degree in optical engineering from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2007, where he is currently pursuing his PhD degree in optical engineering. His current research interests include image restoration, real-time image processing, and general purpose GPU.

Bao Zhang is a professor at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. He received his BS degree in optical instrumentation, MS degree in fine instrumentation, and PhD degree in mechanical engineering from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 1989, 1994, and 2004, respectively. His major research interest is the research and manufacture of opto-electronic reconnaissance platforms. He is a Member of the Chinese Optics Society.

Hui Sun received his BS degree in mathematic computing from Jilin University in 1985. He has been a professor at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. He is devoted to the design and development of computer software. His current research interests include digital image processing, analysis, and computer simulation.