Publ. Astron. Soc. Japan (2020) 72 (2), 30 (1–14)
doi: 10.1093/pasj/psaa006

A low-cost and duplicable portable solar adaptive optics system based on LabVIEW hybrid programming

Deqing REN[1,*] and Gang WANG[2,3,4]

[1]California State University Northridge, Physics and Astronomy Department, 18111 Nordhoff Street, Northridge, CA 91330, USA
[2]National Astronomical Observatories/Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China
[3]CAS Key Laboratory of Astronomical Optics & Technology, Nanjing Institute of Astronomical Optics & Technology, Nanjing 210042, China
[4]University of Chinese Academy of Sciences, Beijing 100049, China

*Email: ren.deqing@csun.edu

Received 2019 July 23; Accepted 2020 January 20

Abstract

We have developed a portable solar adaptive optics (PSAO) for diffraction-limited imaging based on today's multi-core personal computer. Our PSAO software is written in LabVIEW code, which features block-diagram function based programming and can dramatically speed up the software development. The PSAO can achieve a ~1000 Hz open-loop correction speed with a Shack–Hartmann Wave-front Sensor (SH-WFS) in 11 × 11 sub-aperture configuration. The image shift measurements for solar wave-front sensing are the most time-consuming computations in a solar adaptive optics (AO) system. Since our current LabVIEW program does not fully support multi-core techniques for the image shift measurements, it cannot fully take advantage of the multi-core computer's power for parallel computation. In order to accelerate the AO system's running speed, a dedicated message passing interface/open multi-processing parallel programming technique is developed for our LabVIEW-based AO program, which fully supports multi-core parallel computation in LabVIEW programming. Our experiments demonstrate that the hybrid parallel technique can significantly improve the running speed of the solar AO system, and this work paves the way for the applications of a low-cost and duplicable PSAO system for large solar telescopes.

Key words: instrumentation: adaptive optics — instrumentation: high angular resolution — techniques: image processing

1 Introduction

To correct atmospheric turbulence perturbation induced wave-front error, solar adaptive optics (AO) systems have been developed over the past few decades for ground-based diffraction-limited imaging. To ensure high performance for a solar AO system, the real-time measurements of wave-front errors are critical. Compared with night AO systems, solar AO wave-front sensing must work on extended targets such as sunspots, pores, or solar granules. Therefore, solar AO wave-front sensing usually adopts a cross-correlation

algorithm and samples a small field of view as a guide region with at least 20 × 20 pixels, which is extremely time-consuming (Rimmele et al. 2004).

Two kinds of hardware computing platforms are currently used for solar AO systems (Rimmele 2004). The dedicated hardware platform, which is based on Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) is adopted by solar AO systems on the 0.76 m Dunn Solar Telescope (DST) (Langlois et al. 2004), the 26 cm solar fine structure telescope (Rao et al. 2010) and the 1 m New Vacuum Solar Telescope (Kong et al. 2016). Although dedicated hardware platforms are superb for fast calculation of digital image processing in parallel with low latency, it is expensive, with less flexibility to modify the AO configuration for different telescopes. Since dedicated hardware is used, software programming is highly based on the deployed hardware, and knowledge of both hardware and DSP/FPGA is needed in the AO development, which is highly time-consuming.

The other approach uses multi-core based general computers for software programming. This approach has been adopted by the solar AO systems on the Vacuum Tower Telescope (VTT) (van der Luehe et al. 2003), the GREGOR Solar Telescope (Berkefeld et al. 2012), and the Big Bear Solar Telescope (BBST) (Schmidt et al. 2014). Such general computer based programming is more flexible and a large amount of existing software resources can be used. However, current systems use low-level computer programming languages such as C/C++, and are also time-consuming, which is a limiting factor for rapid AO system development at low cost.

In our previous work, we developed a portable solar adaptive optics (PSAO) system (Ren et al. 2010, 2015). Based on general multi-core personal computer and the high-level LabVIEW programming language, this system is developed using off-the-shelf components, including all the electronic and optical components. LabVIEW, developed by National Instruments, is one of best and most popular software packages for electronic hardware device control and programming. Almost all electric hardware devices, such as the deformable mirror, tip-tilt platform, and wave-front sensor camera used in our solar AO, have LabVIEW software drivers or LabVIEW programming example codes provided by the vendors. Further, many LabVIEW standard functions can be used immediately for our AO software programming, and LabVIEW's block-diagram based graphical development environment dramatically reduces the software development time and increases productivity, which makes it suitable for the rapid development of a prototype. The system is compact, with a physical size of 900 mm × 600 mm × 250 mm. Such a portable hardware system and flexible software configuration make it possible for the rapid construction of an AO system for any solar facility telescope at a low cost. Initial on-site testing and a trial observation were conducted on the 0.61 m solar telescope at the San Fernando Observatory (Ren & Dong 2012). Our PSAO is duplicable and was successfully used as a visitor instrument with both the National Solar Observatory's Kitt Peak 1.6 m McMath–Pierce Solar Telescope (McMP) and the Sunspot 0.7 m Dunn Solar Telescope for potential scientific observations (Ren et al. 2010), which fully demonstrated the potential of the portable solar AO technique.

One of the most time-consuming parts in a solar AO system is the measurement of image shifts in wave-front sensing. In our PSAO programming, the LabVIEW's standard function "Pattern Match," instead of the traditional cross-correlation calculations, is used for the measurements of image shift. Based on the pyramidal pattern match algorithm, the "Pattern Match" function is widely used in machine vision and image processing, which can also be used for solar AO wave-front sensing due to its ability to make fast and sub-pixel accurate measurements towards image shift calculation. While LabVIEW codes in general automatically support multi-cores for parallel calculations, LabVIEW's "Pattern Match" functions do not support multiple cores for parallel calculation due to its inherent algorithms used for the pattern match, which seriously limit the correction speeds of the PSAO if a large number of wave-front sub-apertures is used, i.e., for the future large solar telescopes. To overcome this limitation, a hybrid LabVIEW programming technique is adopted, in which an message passing interface/open multi-processing (MPI/OpenMP) parallel technique in C++ programming is used to improve the running speed in image shift calculations of the wave-front sensing, which is further integrated as a sub-function of the LabVIEW code, and thus can be considered by LabvIEW as a user-defined function. In this way, we can continue to enjoy LabVIEW's advances for rapid AO software development, while still delivering high-speed wave-front sensing and calculation.

2 Initial PSAO system and performance

The general principle of the PSAO optics system is shown in the layout in figure 1. The light incident from the telescope focal plane (FP) is collimated by lens L1, and then is reflected by the tip-tilt mirror (TTM) and deformable mirror (DM) subsequently. A beam splitter (BS) divides the light to the Shack–Hartmann wave-front sensor (SH-WFS) and the science camera, respectively. Lens L2 is used to form the solar image on to the science camera with a proper focal ratio. A pair of lenses, L3 and L4, are used to form an image of the telescope aperture on to the microlens array

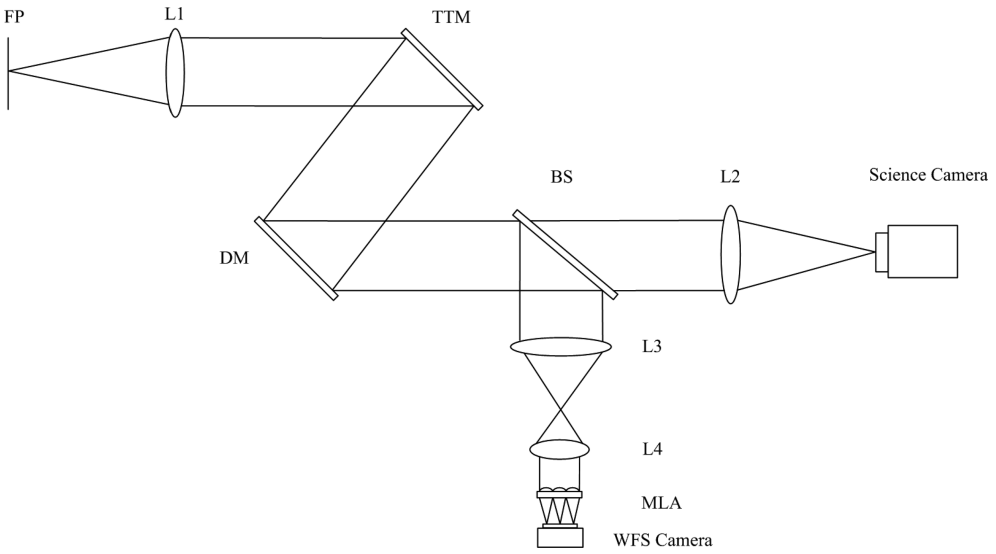

Fig. 1. Schematic diagram of the PSAO optics. In order to keep a compact size, the MLA is directly integrated into the WFS camera.

of the SH-WFS. As a portable AO system, our PASO uses a minimum of optical components. For different telescope, we only need to replace Lenses L1, L2, L3, and L4, which makes our PSAO hardware duplicable for any current solar telescope.

The PSAO uses one multi-core computer to perform real-time computations and control tasks, including capturing the wave-front sensor images, calculating the control signal, and controlling the correction components. In our previous PSAO, all the AO code is written in LabVIEW, which dramatically speeds up the software development. In addition, since our PSAO will be used with different telescopes with different apertures, our LabVIEW based software is flexible, and can use any number of SH-WHS configurations to sample the telescope pupil. For example, it can use 11×11 or 8×8, with or without central obstruction, to sample the telescope's pupil. This makes our PSAO software duplicable to any telescope. All optical lenses and mirrors are based on off-the-shelf components, which allow us to duplicate an AO system for a dedicated telescope quickly at a low cost. Since commercial optical components with moderate wave-front quality are used, possible wave-front errors will be included. In addition, during our observations with a telescope, the PSAO needs to be set up in a few hours, which may inevitably introduce some optical alignment errors. These optical component and alignment errors must be effectively corrected for diffraction-limited imaging with a low-cost and duplicable portable system. These corrections are achieved by using our dedicated focal plane point spread function (PSF) sensing and correction approach (Ren et al. 2012), which directly command the DM; excellent wave-front quality can be achieved in a few tens of minutes of optimization runs that optimize the DM

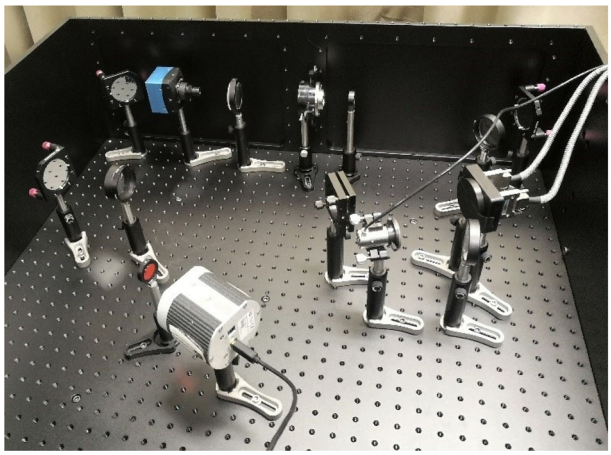

Fig. 2. Photograph of the PSAO system. The MLA and WFS camera are assembled as an integrated unit to keep the system as compact as possible. (Color online)

surface profile to compensate for these wave-front errors. Such a strategy makes it possible to use commercial optical components for our PSAO system. In order to reduce the overall physical size, the system is built to be as compact as possible. For example, the SH-WFS is directly integrated into the wave-front sensor camera. All AO components are installed in an enclosed box with a size of the order of 900 mm × 600 mm × 250 mm, as shown in figure 2. The small physical size allows us to bring it to any solar telescope for scientific observations.

On 2013 October 2, the portable AO was tested with the NSO 1.6 m McMP at Kitt Peak for sunspot observations at the NIR *H*-band. During our observation run, the seeing is in the average local condition (i.e., $r_0 = 5$ cm), with an average wind speed between 10–20mph. In this test, the WFS uses 9×9 sub-apertures, and the PSAO

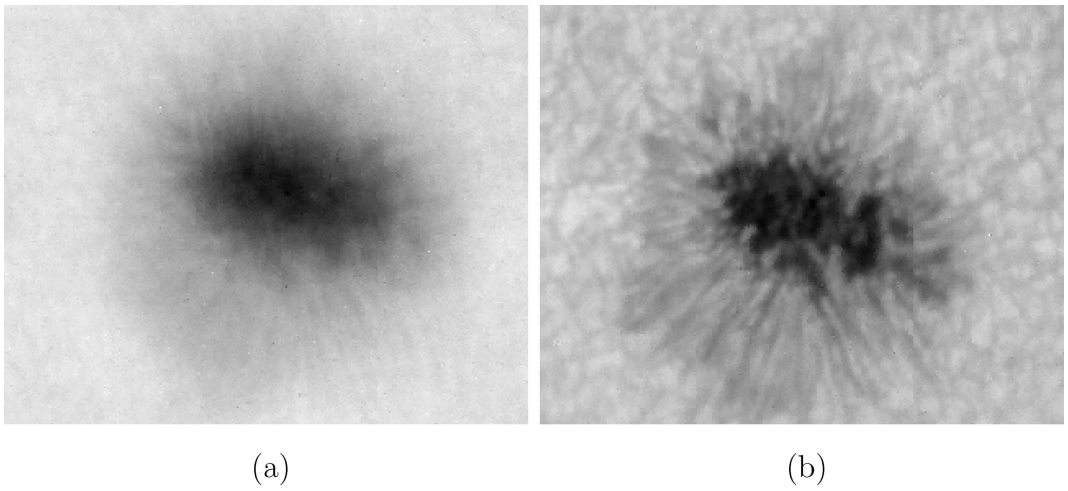

Fig. 3. Sunspot images without (a) and with (b) AO correction, at the McMP.

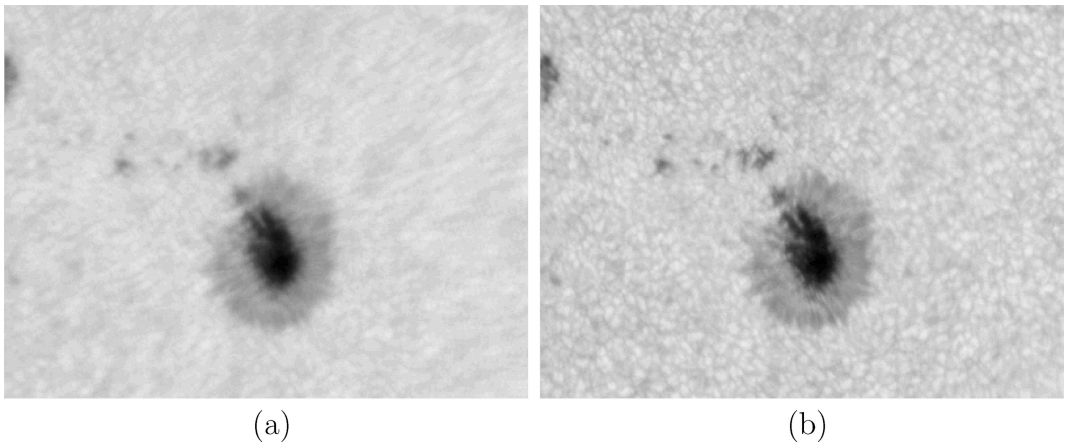

Fig. 4. Sunspot images with AO off (a) and AO on (b) at the 0.6 μm visible wavelength, with the DST.

system adopted a full LabVIEW programming approach and achieved a correction speed of up to 900 Hz (open loop), with a personal computer with an i7 8-core CPU running at 3 GHz clock frequency. A DM purchased from Alpao with 69-actuator was used. The entire sunspot with a field of view (FOV) of $30'' \times 30''$ is used as a guide region for wave-front sensing, which is sampled by 40×40 pixels in each sub-aperture. Figure 3 shows the typical sunspot *H*-band images without and with our AO corrections, respectively. One can see from the sunspot image without AO correction that the average daytime seeing condition at the Kitt Peak is poor, and the sunspot image is seriously distorted by the atmospheric turbulence. However, our AO can still provide effective correction, which can be seen clearly from the AO-corrected image (figure 3b).

After the Kitt Peak AO run, the PSAO 69-actuator DM was updated with a new Alpao DM with a 97-actuator, and was tested with the NSO Sacramento Peak 0.76 m Dunn Solar Telescope (DST), in 2015 June. With the 0.76 m DST, the PSAO used 97 sub-apertures in an 11×11 configuration for wave-front sensing. The open-loop speed of PSAO achieved ~1000 Hz, while the pure speed of wave-front slope measurement was ~1300 Hz. 60-order Zernike polynomials are used to correct the atmospheric turbulence induced aberrations. Since the seeing condition at Sacramento Peak ($r_0 = 7$ cm) is much better than that at Kitt Peak, the portable AO should deliver better correction results, which is confirmed by our recent observation. Figure 4 shows the typical sunspot images with the AO off (left-hand panel) and AO on (right-hand panel) in this observation run with the DST. Again, the portable AO uses a $30'' \times 30''$ FOV for wave-front sensing that encloses the entire sunspot, and thus our AO actually works as a ground layer AO (GLAO) system, so that it can effectively correct the ground-layer turbulence at good seeing conditions and achieve a large FOV correction. The images in this figure

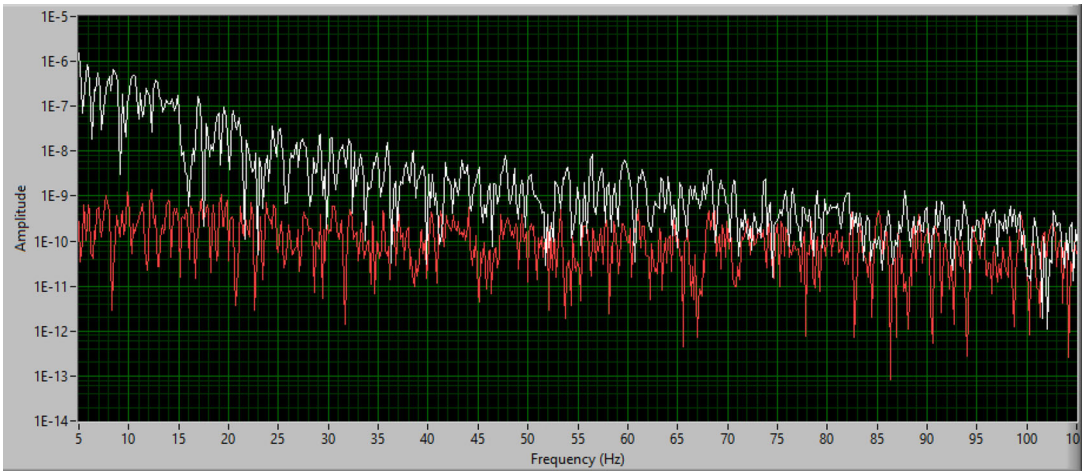

Fig. 5. PSD of the PSAO. Red and white lines represent AO on and off, respectively. (Color online)

have a FOV of $\sim 50'' \times 50''$. It is clear that at a site with a good seeing conditions, our portable AO can deliver a good performance over a large FOV.

Power spectral density (PSD) is used to characterize the efficiency of the AO correction speed. We use Zernike polynomials to calculate the PSD. For N terms of Zernike polynomials that are used to characterize the wave-front errors, to calculate its PSD we first calculate its root mean square (rms) as

$$a_{\rm rms} = \sqrt{a_2^2 + a_3^2 + \cdots + a_N^2}, \tag{1}$$

where a_i is the i-term coefficient of the Zernike polynomials. The piston term a_1 is excluded from the above calculation. The rms value of N terms of coefficients is used to estimate the PSD. In general, we use 5000 images, in which 60 orders of Zernike polynomials are used to calculate the rms value in each image, to estimate the distribution of the wave-front power as a function of the frequency. Based on the wave-front rms value, a dedicated LabVIEW code was developed to calculate the PSD, in which a standard LabVIEW function for the PSD calculation was adopted. Figure 5 shows the PSD of our PSAO, in which the red line represents the PSD with AO correction, while the white line denotes that without AO correction, which is also the original PSD of the atmospheric turbulence. The two lines join at 80 Hz, indicating that our PSAO has a bandwidth of 80 Hz.

The PSAO static aberrations were corrected by using the science camera to measure and optimize the focal plane PSF and directly command the DM to correct the wave-front aberration of the AO system. This technique is delicately developed to correct the AO static and non-common optical wave-front error (Ren et al. 2012), which allows us to use low-cost, off-the-self optical components to construct our PSAO, since any potential wave-front error induced by these components will be corrected by the DM. Since all the static and non-common associated errors are effectively corrected, we only consider the remaining dominant wave-front errors for our PSAO performance estimation. The overall residual variance for the solar AO can be estimated from the remaining three dominant errors;

$$\sigma^2 = \sigma_{\rm fit}^2 + \sigma_{\rm bw}^2 + \sigma_{\rm other}^2, \tag{2}$$

where $\sigma_{\rm fit}^2$ is the fitting error that represents the wave-front variance due to the limited DM actuator number. For a DM sample spacing $r_{\rm s}$ and seeing r_0, it can be calculated as $\sigma_{\rm fit}^2 = 0.349(r_{\rm s}/r_0)^{5/3}$ (Tyson 2010). $\sigma_{\rm bw}^2$ is the lag error due to finite AO bandwidth. $\sigma_{\rm other}^2$ represents all the other remaining errors. The lag error is determined by the bandwidth and the average wind speed (which is $v = 7\,{\rm m\,s^{-1}}$ for our case). According to Tyson (2010), $\sigma_{\rm bw}^2 = (f_G/f)^{5/3}$, where for a wind speed v and seeing parameter r_0, $f_G = 0.43(v/r_0)$. For a bandwidth of 80 Hz and seeing parameter $r_0 = 7$ cm at the Sunspot, it yields a lag error of 0.35 rad^2 variance.

Since enough photons are available for solar wave-front sensing, readout and photon noises are not included. As we limit our calculation for a small imaging angle, isoplanatic angular error is excluded, although we regularly use a large wave-front sensing FOV. For $\sigma_{\rm other}^2$, we consider the wave-front sensing correlation error (Michau et al. 1993), which has a typical value of 1/17 waves (Ren & Zhao 2016). Based on above discussions, we have the overall residual variance according to equation (2) of

$$\sigma^2 = 0.30 + 0.35 + 0.14 = 0.79\ {\rm rad}^2 \tag{3}$$

For small residual variance, the Strehl ratio can be calculated by using the analytical equation,

$$S = e^{-\sigma^2} \tag{4}$$

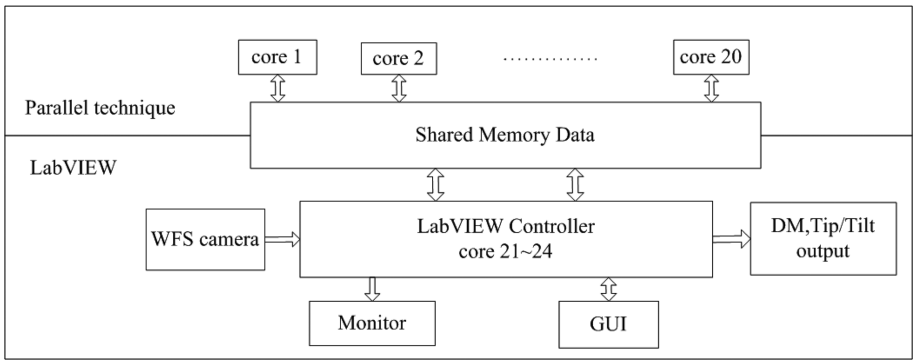

Fig. 6. Schematic diagram of the upgraded PSAO control software.

Applying this to our PSAO, the Strehl ratio is 0.45, which is surprisingly lower than what we estimated. However, this is a typical performance for solar AO systems. It is obvious that our PSAO is almost equally dominated by the errors of DM fitting (0.30 rad^2 variance) and bandwidth (0.35 rad^2 variance). The DM fitting error can be reduced by using more actuators, which will result in an increase of the wave-front sensor sub-apertures, and thus needs fast computations. To reduce these two types of dominant errors, a high speed of AO wave-front sensing and calculation is critically needed.

LabVIEW's pattern matching (Kwon & Ready 2014) is based on a pyramidal matching algorithm, and it uses normalized cross-correlation as a core technique. This pattern matching process consists of two stages: learning and matching. During the learning stage, the algorithm extracts gray value information from the template image. The algorithm organizes and stores the information in a manner that facilitates faster searching in the inspection image. The information learned during this stage is stored as part of the template image. During the matching stage, the pattern matching algorithm extracts gray value information from the inspection image. Then, the algorithm finds matches by locating regions in the inspection image where the highest cross-correlation is observed. In principle, LabVIEW's block-diagram based codes automatically support multi-cores for parallel calculations. However, because of the nature of LabVIEW's Pattern Match algorithm, there are two issues which limits its applications for AO solar wave-front sensing. First, it can only use an image with at least 31×31 pixels, since a small size of sampled image of less than 31×31 pixels would not be able to provide enough features for the pattern match operation. For the solar guide region, a FOV on the order of $10'' \times 10''$ is generally adopted, which could be sampled by 20×20 pixels. Therefore, the 31×31 pixel sampling will significantly reduce the image shift measurement speed, compared with other AO systems that use 20×20 pixels. Secondly, we found that LabVIEW's Pattern Match function cannot fully support multi-cores for parallel computation, which seriously limits its applications for solar AO systems with large telescopes.

3 LabVIEW hybrid AO programming

In order to overcome LabVIEW's Pattern Match inherent limit and to achieve a high speed of image shift measurement, a hybrid approach that integrates a LabVIEW block-diagram based program with the high-speed parallel techniques of MPI and OpenMP is developed to upgrade the PSAO system, with the controlling diagram shown in figure 6. The parallel calculation is based on C++ code, implemented via shared memory technology (Bershad et al. 1991). The MPI/OpenMP parallel code is integrated into the original PASO LabVIEW code via LabVIEW's dynamic linked library. The shared memory allows the data to directly operate the system memory at high speed using multi-cores, which has a great advantage for large data transfers. By performing reading and writing operations on shared memory DLL, the C++ codes read the wave-front sensor image and the image shift calculation results can be transferred back to the LabVIEW code. As demonstrated in the following section, the hybrid programming technique can fully use the CPU resources of multi-cores for parallel computation. The multi-core computer for testing the hybrid parallel technique has two Intel Gold 6136 CPUs and each CPU has 12 cores running at 3 GHz clock frequency. As shown in figure 6, 20 cores are used simultaneously for parallel image shift measurement, while the remaining four cores are automatically assigned for other tasks, such as the operating system and hardware control, and so on.

The MPI (Gropp et al. 2000) and OpenMP (Chapman et al. 2008) techniques used in our programming are two open-source Application Program Interfaces (APIs).

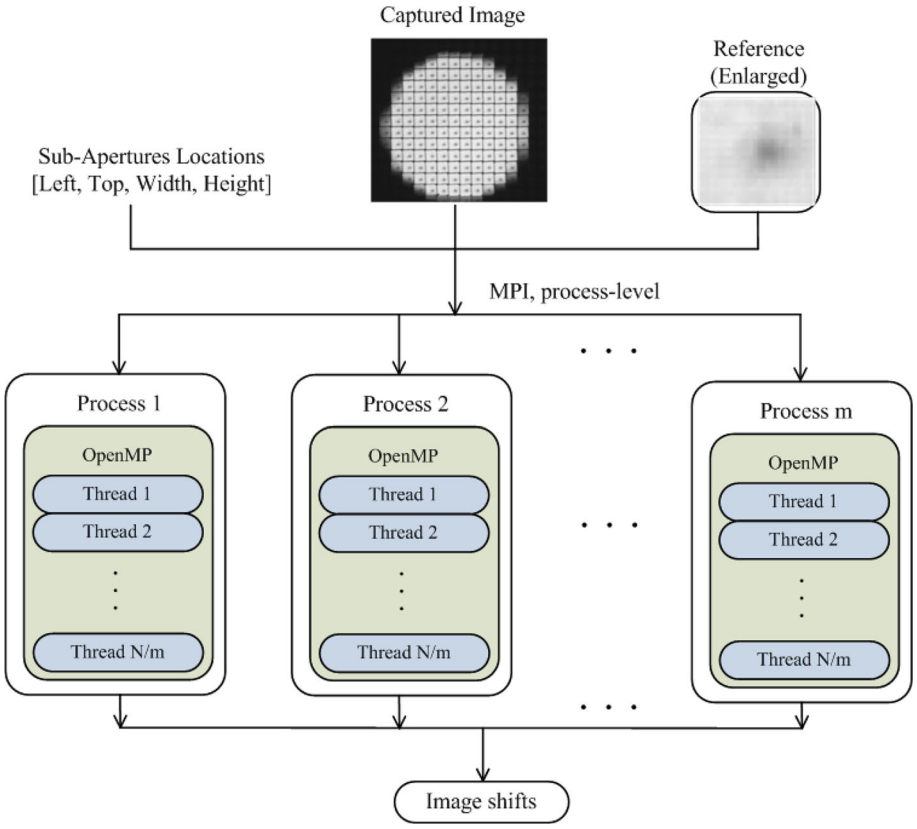

Fig. 7. Dedicated MPI/OpenMP parallel technique developed for the PSAO wave-front sensor. (Color online)

Both APIs are widely used in the field of parallel calculation because of their good performance (Smith & Bull 2001). The parallel techniques are hierarchical, with MPI parallelism in the upper layer, and OpenMP parallelism in the lower level, providing lightweight parallelism within each MPI process. The communication in each MPI process is achieved through a message-passing mechanism, which enforces data synchronization by the point-to-point block sending and receiving message functions called "*MPI_Send*" and "*MPI_Recv*" to ensure valid data access. Conversely, the communication of OpenMP in thread-level is implemented through access to each process's shared memory, and appropriate synchronization is used to ensure that the real-time wave-front sensor data are calculated and then used.

As the image shift for each sub-aperture can be measured and calculated individually, a dedicated MPI/OpenMP parallel technique is developed to fully support the parallel calculation for our PSAO wave-front sensor. As shown in figure 7, all of the MPI lies outside the OpenMP parallel construct. In each MPI process, the OpenMP directives for the thread creation are inside the loop. Suppose the multicore computer consists of N cores and there are M efficient sub-apertures, the parallel program starts by loading the non-real-time AO calibration data, and then the workflow is executed via the following procedure:

(1) Divide the sub-aperture images into m parts, with each part containing M/m sub-aperture images;
(2) Send each part of sub-aperture images and the reference image to each process;
(3) Set N/m threads for OpenMP to parallel the measurements of image-shift in each process and measure image shifts of M/N sub-aperture images with sub-pixel cross-correlation accuracy in each thread;
(4) Collect measurement results and send them to one of the processes.

4 Image shift measurements and performance

The image shifts are estimated by measuring the displacements of solar structure features between each sub-aperture and the reference images. In this paper, the squared difference function (SDF), one of the cross-correlation algorithms, is used to optimize the image shift measurement in each sub-aperture. As discussed by Lofdahl, the SDF algorithm performs better among the various cross-correlation algorithms (Löfdahl 2010).

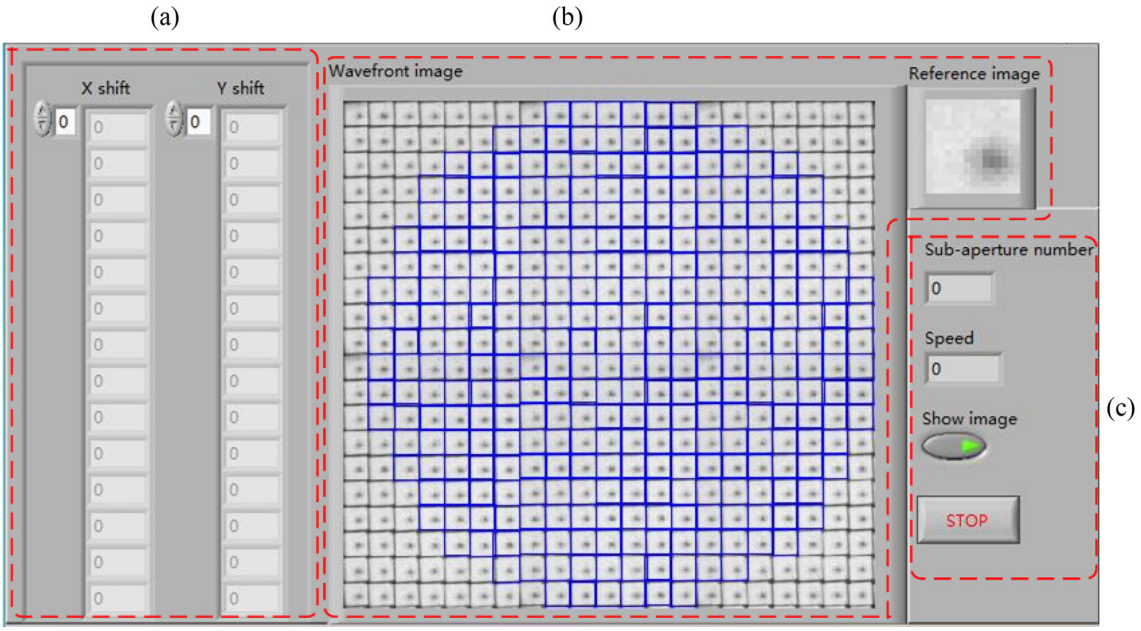

Fig. 8. GUI of the simulation LabVIEW software, in which 316 sub-apertures in a 20 × 20 WFS configuration are used for the evaluation. (Color online)

The SDF calculates the square difference for each pixel between the reference and sub-aperture images with pixel-level accuracy. Assuming that the sub-aperture and reference images have $M \times N$ pixels and $m \times n$ pixels ($M > m$, $N > n$), respectively, the correlation function $\mathbf{C}(i, j)$ of sub-aperture g and reference g_{ref} is calculated as

$$\mathbf{C}(i, j) = \sum_{x,y} [g_{\mathrm{ref}}(x, y) - g(x, y)]^2. \quad (5)$$

The location of minimum C_{min} in matrix **C** yields the image shift position, which is the result of the pixel-level matching. We set the coordinate of C_{min} as the center, and select a window from matrix **C**. By applying a bilinear interpolation algorithm to resize the window by N times, the minimum element with sub-pixel accuracy can be achieved. The above algorithms are developed using the open-source computer vision library, known as OpenCV (Bradski & Kaehler 2008).

In order to evaluate the performance of the parallel calculations, we have developed a simulation software package based on LabVIEW programming. Figure 8 shows the main interface window of the simulation software, which consist of three sections. The image shifts in X and Y directions are shown in the section (a), and can be saved in a text file; in the middle of section (b) in the panel, a WFS image and reference image are displayed. The blue grids represent the effective or selected sub-apertures for the wave-front sensing measurement; section (c) shows the number of valid sub-apertures and image shift measurement speed. In this evaluation work, we use 316 sub-apertures in a 20 × 20 WFS configuration, as shown in figure 8.

Table 1. Sub-pixel accuracy results of the parallel approach and the LabVIEW Pattern Match.

	Resize	X (pixels)	Y (pixels)
Parallel approach	2	0.254	0.249
	4	0.131	0.147
	8	0.185	0.189
Pattern Match function	—	0.118	0.106

Since LabVIEW Pattern Match has been successfully used in our PSAO system for the image-shift measurements and diffraction-limited imaging, we therefore implement it as a standard to evaluate the speed and accuracy of the parallel technique. The sub-aperture images have the sizes of 31 × 31 and 24 × 24 pixels, respectively. The reference image, which is part of one sub-aperture, is also defined to accommodate a shift range up to ±4 pixels in each direction. An off-the-shelf multi-core computer, which has two Intel Xeon 6136 CPUs (each CPU consists of 12 cores at 3.00 GHz), is used for this calculation.

The calculation accuracy of the parallel algorithm is established by comparing the absolute difference between measured image shifts and real image shifts. We firstly generate a high-resolution image, shift it by a known pixel, and then down-sample it to the sub-aperture image scale. The average error in each direction is used to quantify this accuracy. The errors obtained by the parallel approach and Pattern Match are listed in table 1. As shown, in general, increasing the pixel sampling size in the parallel algorithm will deliver better accuracy. When a four-time resize factor is adopted, the accuracy of parallel algorithm will

Table 2. Possible process/thread allocations for the hybrid parallel techniques and LabVIEW Pattern Match.

	N_{CORE}	N_{PROCESS}	N_{THREAD}
Parallel 1 (two processes, 10 threads)	20	2	10
Parallel 2 (four processes, five threads)	20	4	5
LabVIEW Pattern Match	20	—	—

achieve 0.131 and 0.147 pixels in the X and Y directions, respectively, which is comparable to the Pattern Match developed by the National Instruments (NI). Therefore, the four-times resize factor is used for our parallel algorithm.

Matching a reference to sub-apertures is an iterative procedure and can be implemented with a *for Loop*. In our previous works, a LabVIEW parallel model is adopted and *for Loop* execution is accelerated by enabling parallel for loop iterations. In this paper, the *for Loop* execution is accelerated by a hybrid parallel technique. For the hybrid parallel technique, given the same number of computer cores (N_{CORE}), the more MPI processes (N_{PROCESS}) we choose, the fewer cores available that OpenMP threads (N_{THREAD}) could use. To gain the best performance for the hybrid parallel technique, we need to analyze the most suitable numbers of MPI processes and OpenMP threads. As an optimal result, 20 cores (N_{CORE}) are used for parallel calculation. Two good parallel technique models with different processes and threads, namely Parallel 1 (two processes, 10 threads) and Parallel 2 (four processes, five threads), as well as a LabVIEW parallel model, are list in table 2.

To evaluate the performance of the hybrid parallel technique, the running speed of the hybrid parallel technique (hybrid parallel model with SDF algorithm) and LabVIEW Pattern Match (LabVIEW parallel model with Pattern Match algorithm) are compared. We implement the resolution of 31 × 31 and 24 × 24 pixels for each sub-aperture for this comparison. The 31 × 31 pixels sub-aperture sampling was used for our previous PSAO system to measure the image shifts. We can take it as a standard to evaluate the speed of the hybrid parallel technique. Figure 9 shows the performance of the hybrid parallel technique in the case of the 31 × 31 pixels sub-aperture sampling. As shown, the running speeds is 2.4 times that of LabVIEW Pattern Match, at different sub-aperture numbers. Please note that the PPM is based on a fast pattern search technique and is much faster than that of the SDF in the case of a single CPU-core, which was confirmed by our comparison of these two algorithms with a LabVIEW code. However, our hybrid parallel programming is still faster than that of the LabVIEW native program, because of the employment of the fully-parallel calculation. The above result proves that the hybrid technique is more efficient than the LabVIEW Pattern Match.

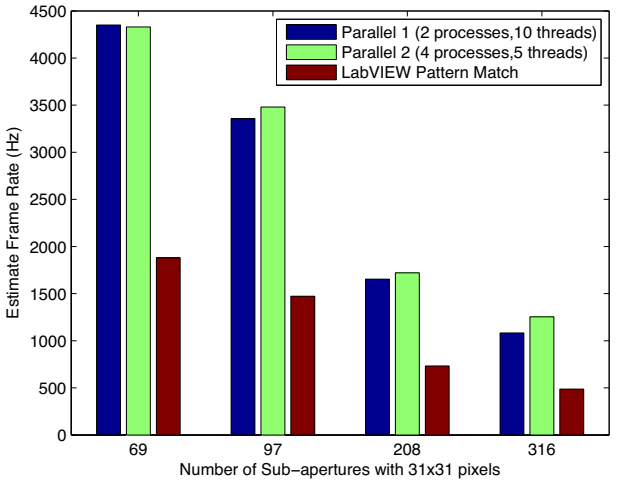

Fig. 9. Running speeds of the LabVIEW Pattern Match and the parallel technique in the 31 × 31 pixel sub-aperture case. (Color online)

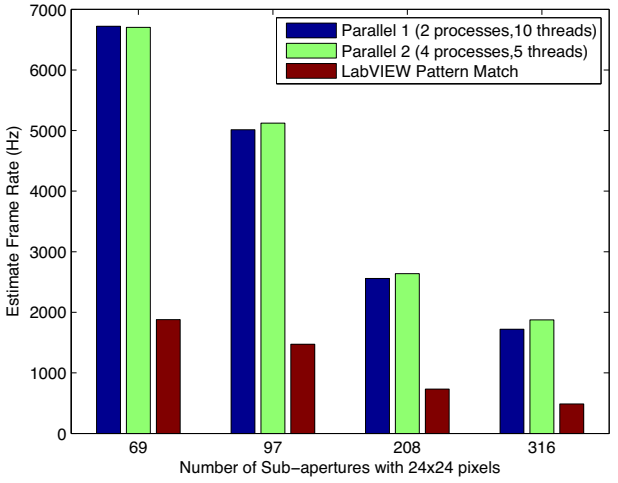

Fig. 10. Running speeds of the LabVIEW Pattern Match and the parallel technique in the 24 × 24 pixel sub-aperture case. (Color online)

Figure 10 shows the performance of the hybrid parallel technique in the case of the 24 × 24 pixels sub-aperture sampling. The hybrid parallel technique outperforms the LabVIEW Pattern Match, with a running speed of 3–3.3 times that of the LabVIEW Pattern Match, which proves the excellent performance of the parallel approach for a small size sub-aperture sampling. When the number of sub-apertures is set to 316, the running speed of the hybrid parallel technique achieves 1.9 kHz, which is comparable to the BBSO's high-order solar AO (T. Berkefeld & S. Dirk 2012).[1] This result clearly indicates that a high-efficiency image-shift measurement technique is achieved for our LabVIEW hybrid programming. The resolution of the 24 × 24 pixels per sub-aperture is able to track solar fine structures and thus can be used for our future solar AO systems.

[1] Berkefeld, T., & Dirk, S. 2012, Paper presentation, Real Time Control for Adaptive Optics Workshop, ESO (Garching, Germany).

Table 3. AO system running speeds at 24 × 24 pixels per sub-aperture.

Number of sub-apertures	97(11 × 11)	208(16 × 16)	316(20 × 20)
AO running speed (Hz)	3366(4397)	1796(2285)	1225(1557)

Finally, we integrated the above hybrid parallel technique software package into our PASO system and measured the AO system running speed sampled at 24 × 24 pixels per sub-aperture at different WFS configurations of 11 × 11, 16 × 16, and 20 × 20, with the results as shown in table 3. In addition to showing the running speed of entire AO system (the values without parentheses), we also show the AO running speed excluding the WFS camera (the values within parentheses). In this experiment, the EoSens 3CXP camera is used as the WFS camera. The EoSens 3CXP is a CoaXPress camera, and has a running speed of 566 frames per second at 1696 × 1710 pixels resolution. In our test, it delivers a running speed of 14358, 8394, and 5755 Hz for the 97, 208, and 316 WFS sup-apertures, respectively. It is clear that the entire AO system can deliver a running speed well above kilohertz, at different WFS configurations, although this speed is reduced somehow by the limited imaging grab speed of the WFS camera.

5 Laboratory test and result

In contrast to the nighttime astronomy that uses point source stars for wave-front sensing, the scientific targets for the solar AO systems are two-dimensional solar surface structures, such as sunspots, flares, and granules. Therefore, solar adaptive must use the two-dimensional structure for wave-front sensing in observations . While the nighttime AO can use a point source star for performance evaluation, such as to calculate the Strehl ratio (SR), which is one of the most important steps in characterizing the performance of an AO system (Tyson 2015), the characterization and performance evaluation via the two-dimensional observational structures have proved to be difficult and challenging. To address this issue, we have developed a general platform of atmospheric turbulence perturbation simulation in our laboratory, which can generate the turbulence perturbation corresponding to different seeing conditions and different wind speeds. The simulator uses a point source and thus allows us to directly calculate the Strehl ratio to fully characterize our solar AO performance.

Traditionally, atmospheric turbulence simulators are widely used to characterize the closed-loop performance for nighttime AO systems. The most frequent approach is to use a phase plate (Thomas 2004; Lee et al. 2006). One rotates it to generate time-varying wave-front aberrations. However, it remains difficult to manufacture the phase plate to have a random phase but with certain required spatial frequency components, e.g., the Kolmogorov power spectrum. Another approach is to use a liquid-crystal spatial light modulator (Giles et al. 2000; Taylor & Gregory 2002; Hu et al. 2006). Compared to the static phase plates, this type of modulator can produce a dynamic turbulence wave-front, but it has a relatively slow temporal response time, of the order of up to a few tens of hertz, compared to that provided by a deformable mirror. In addition, it requires the use of polarized light and offers only moderate light transmission. Atmospheric turbulence simulators generated by a DM were also reported (Lee et al. 2017). Here, we use a DM to generate the turbulence perturbation, since the DM can generate a phase screen over a speed faster than 1000 Hz, which will be enough to characterize our solar AO system. According to our knowledge, it will be the first time that a DM based atmospheric turbulence simulator has be used for solar AO system performance characterization, and such a system is highly needed for solar high-angular resolution imaging.

Figure 11a shows the phase screen image of 10 × 10 m^2 size that will be used for our turbulence simulator. The phase screen is generated by the using the end-to-end simulation software package YAO (R. François & V. D.

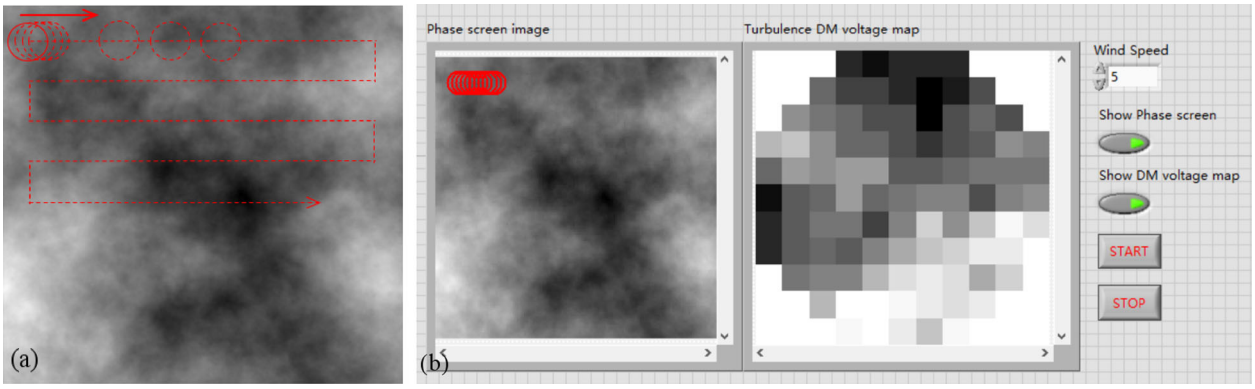

Fig. 11. Turbulence phase screen and the software GUI of the simulator platform. (Color online)

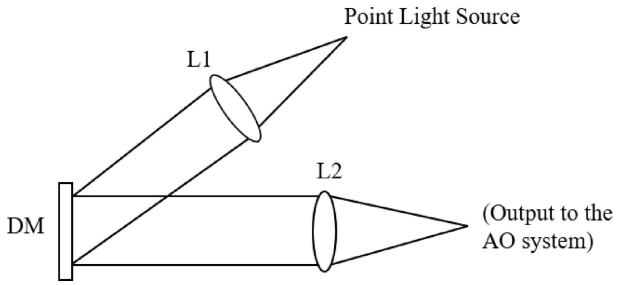

Fig. 12. Schematic diagram of the atmosphere turbulence simulator platform.

Marcos 2013).[2] We used YAO software to generate a phase screen with $10 \times 10\,\mathrm{m}^2$ size and a Fried parameter of $r_0 = 7\,\mathrm{cm}$, which corresponds to a typical solar seeing condition at a solar observatory. In general, one can generate any size of phase screen with any Fried parameter value. The circle represents a 1 m telescope aperture moving along the large phase screen at different instant moments, and the arrow and the path indicate the telescope's direction of movement. In this figure, the telescope samples the phase screen from left to right and then from top to bottom, and this completes one cycle. In our test, the simulator repeats this cycle in an itineration loop. Please note that moving the telescope at a speed on the phase screen is equivalent to moving the phase screen, and thus by adjusting the telescope moving speed we can simulate the wind at a specific speed. All the software associated with the simulator was written using LabVIEW codes. Figure 11b shows the GUI of our simulator. Figure 12 is the schematic diagram of the our simulator system. The light from a point source optical fiber is collimated and reflected by a DM that is used to generate the required phase errors of the Kolmogorov power spectrum. The simulator output is used to as the input to our solar AO system that is being tested. Figure 13 shows the actual experimental system of the simulator in the laboratory, in which a DM purchased from Aplao with 97 actuators is employed. To fold the light and make the system compact, we added a few more flat mirrors to the actual simulator platform.

Our portable solar AO is constructed according the schematic diagram shown in figure 1. The tip-tilt mirror (TTM) was provided by PI Corporation (PI S-330.2SH). The DM was purchased from the ALPAO corporation and had 88 actuators (in 10×10 configuration, excluding those actuators in the four corners). A high-speed camera, the EoSens 3CXP, was purchased from Mikrotron GmbH and used as the WFS camera. For our portable AO system, one of the limiting factors in its imaging performance is the Non-Common Path Aberration (NCPA) that is introduced by the physical separation between the AO WFS path and the science camera path. In addition, the AO Optical Residual Static Aberration (ORSA), which may be induced by the optical components and the misalignment of the system, may also limit the imaging performance. As such, both the NCPA and ORSA must be effectively corrected in advance, before the AO is put into operation. In this experiment, the NCPA and ORSA of our AO system are corrected by the focal-plane wave-front sensing and correction technique that we proposed previously (Ren et al. 2012), which allows us to correct both the NCPA and ORSA static aberrations in our AO system quickly. Figure 14a shows the initial PSF of our AO system before the NCPA and ORSA correction. As shown, the initial focal-plane PSF has a large static wave-front error. Figure 14b shows the PSF of our AO system after the NCPA and ORSA are corrected for using our focal-plane wave-front sensing and correction technique. It is clear that the AO inherent static wave-front error is dramatically reduced after such correction, and a perfect PSF is delivered. The corrected PSF in figure 14b is defined as the perfect PSF and is used for the calculation of the Strehl ratio (SR).

After the static NCPA and ORSA errors are corrected, we conduct a performance evaluation on our solar AO system. Due to the limitation of the first resonance of the membrane, our current DM88 can only run at the speed of ~1200 Hz. Therefore, to evaluate the performance of our AO system fully, the 20×20 sub-aperture (316 efficient sub-apertures) is used to sample the wave-front. In this test, thanks to the LabVIEW Hybrid Programming, our AO system is able to run at a speed of ~1000 Hz. We first run our AO system without any aberrations. The AO system achieves a stable correction and obtains a perfect PSF image, since all the static aberrations are corrected, as shown in figure 15. Then, we apply the phase screen aberrations with wind speeds of $5\,\mathrm{m\,s^{-1}}$, $10\,\mathrm{m\,s^{-1}}$, $15\,\mathrm{m\,s^{-1}}$, and $25\,\mathrm{m\,s^{-1}}$, respectively, via our atmospheric turbulence simulator platform. Figure 16 shows the focal-plane PSF image without and with the AO closed-loop correction. To calculate the SR, we average 100 images, in each case. Figure 17 presents the measured SR with the AO closed-loop correction with an average seeing condition $r_0 = 7$ cm, at different wind speeds. It is clear that the PSF image quality becomes worse as the wind speed increases. When the wind speed is set to $25\,\mathrm{m\,s^{-1}}$, our AO system can still provide an reliable correction, however, with the SR reduced to 0.65, compared to a value of 0.9 at the $10\,\mathrm{m\,s^{-1}}$ wind speed. In this experiment, AO is running at a speed of ~1000 Hz, which is consentient with our previous estimation listed in table 3. The experiment result shows that our LabVIEW Hybrid

[2] François, R., & Marcos, V. D. 2013, Paper presention, Third AO4ELT Conference (Florence, Italy).

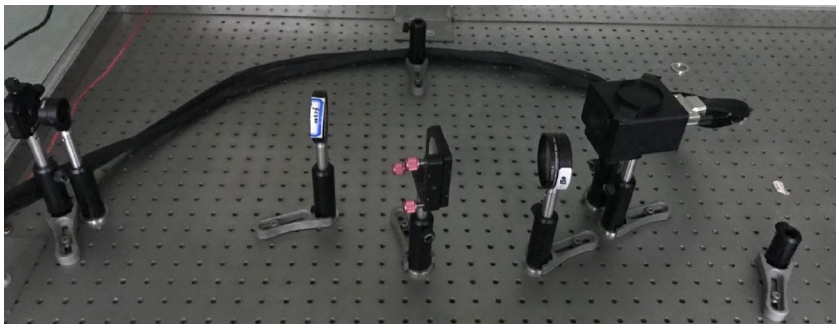

Fig. 13. Actual atmosphere turbulence simulator platform in the laboratory test. (Color online)

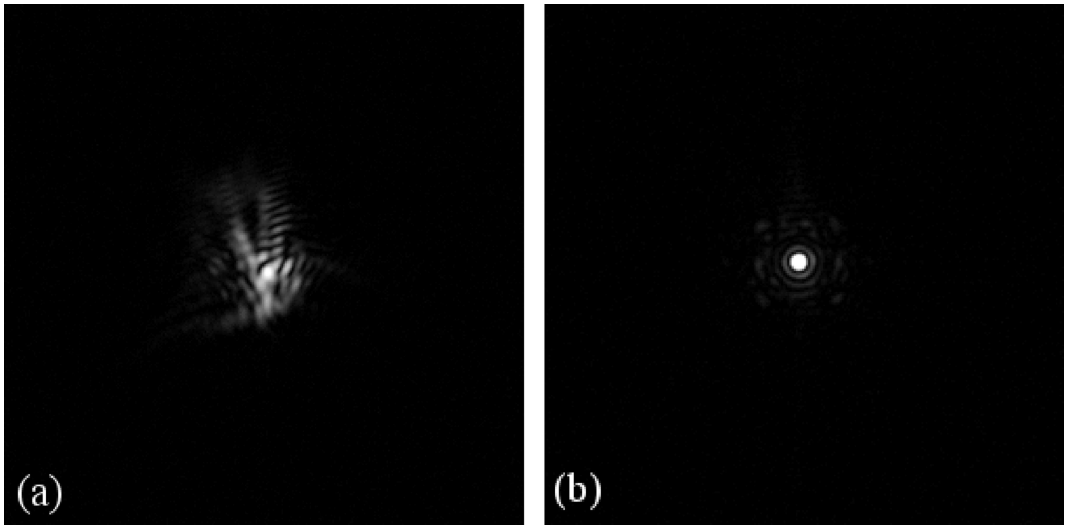

Fig. 14. Focal plane PSF images without (a) and with (b) AO NCPA and ORSA correction, respectively.

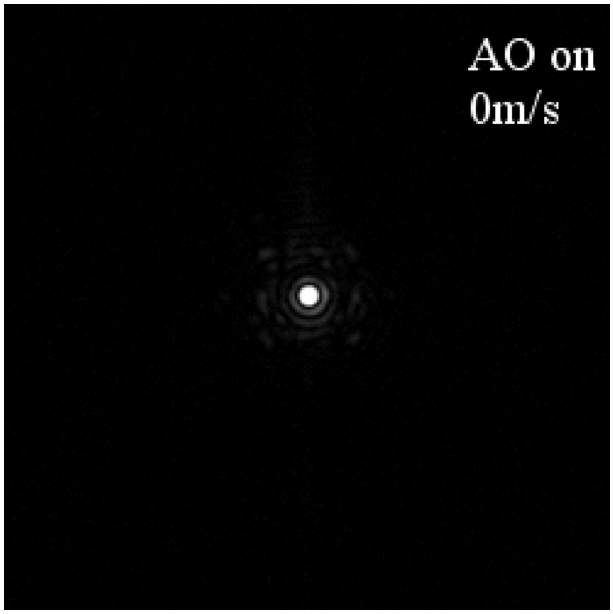

Fig. 15. Focal plane PSF images with the AO NCPA and ORSA correction. The wind speed is 0 m s^{-1} without applying the phase screen aberration.

Programming technique can significantly improve the AO running speed.

6 Conclusions

In this publication, we proposed the first dedicated LabVIEW based hybrid programming technique, which overcomes the LabVIEW limitation and allows parallel calculation for the cross-correlation based image shift measurement for solar AO wave-front sensing. The code for the parallel calculation is written in C++ and can be considered by the LabVIEW code as a user-defined function. Compared with the native LabVIEW Pattern Match, the running speed of the hybrid program is 3–3.3 times quicker, and is comparable with that of other state-of-the-art solar AO systems. To characterize the performance of our hybrid programming AO system fully, we constructed, for the first time, a high-speed atmospheric turbulence simulator platform that is dedicated to the solar AO systems. With the employment of the high-speed solar simulator platform, solar AO performance at different wind speeds and seeing conditions

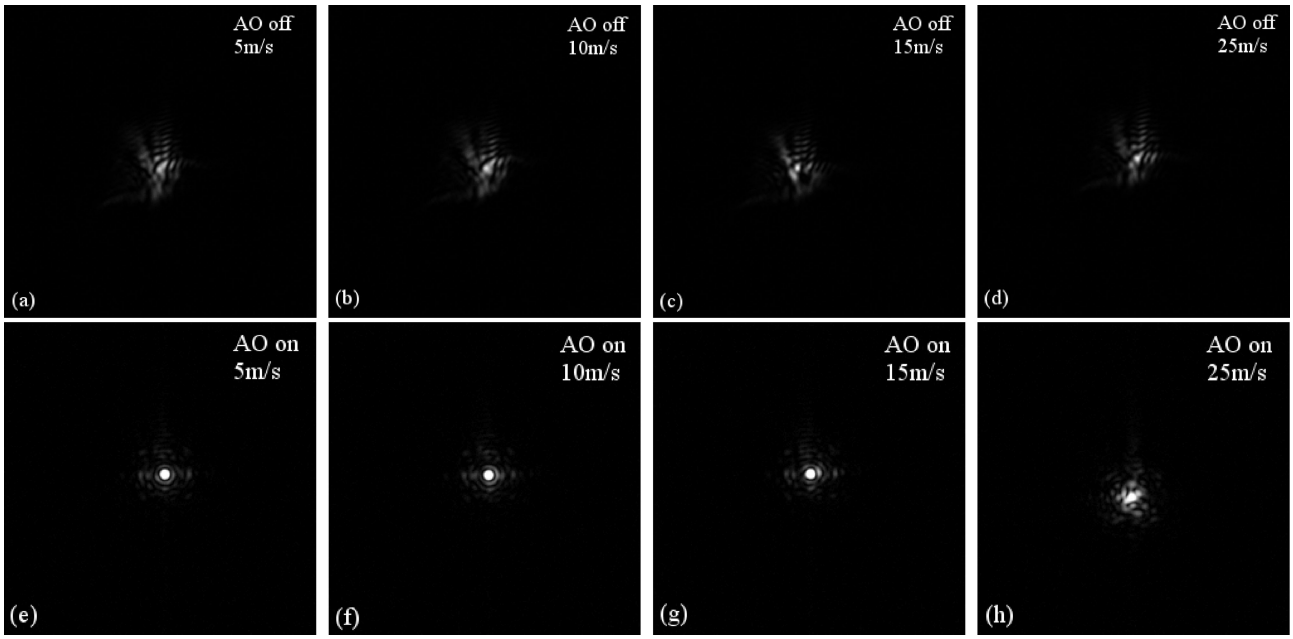

Fig. 16. Focal-plane PSF images without [panels (a)–(d)] and with [panels (e)–(h)] the AO closed loop correction under the wind speed of 5 m s^{-1}, 10 m s^{-1}, 15 m s^{-1}, and 25 m s^{-1}, respectively.

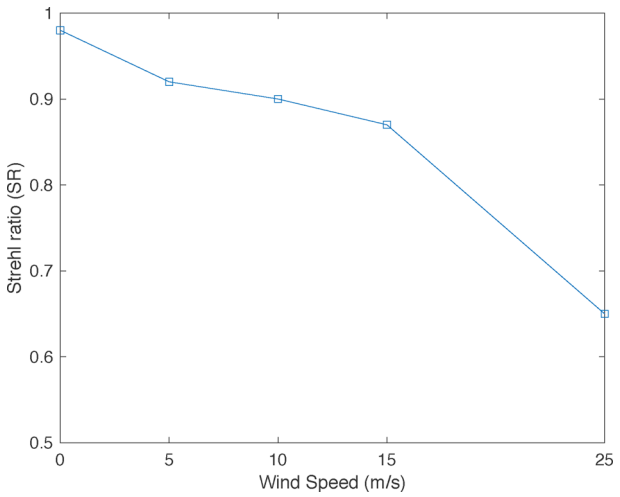

Fig. 17. AO performance measurement results at different wind speeds. "0" means that the wind speed is 0 m s^{-1} without applying the phase screen aberration. "5", "10", "15" and "25" mean that the wind speed is 5 m s^{-1}, 10 m s^{-1}, 15 m s^{-1}, and 25 m s^{-1}, respectively, with the application of the phase screen aberration. (Color online)

can be fully and directly characterized via the measured Strehl ratio. The parallel technique can be used to accelerate image shift measurements and is more efficient than LabVIEW's standard Pattern Match functions, because it makes full use of the available multi-cores for the calculation. The hybrid programming technique allows us to enjoy LabVIEW's unique features for rapid AO software development, which dramatically saves scientists' and engineers' time so that they can concentrate on creative research work such as the development of the novel solar Multi-Conjugate Adaptive Optics (Ren et al. 2015)—a methodology that has not been fully realized for routine operations. This work paves the way to use a low-cost, quickly replicable and portable AO system for future large solar telescopes.

Acknowledgments

We thank the anonymous referee for valuable and thoughtful comments, which significantly improve the quality of thismanuscript. D. Ren acknowledges the supports from National Science Foundation(NSF) under the grants of AST-1607921 and AST-1906166.

References

Berkefeld, T., et al. 2012, Astron. Nachr., 333, 863
Bershad, B. N., Anderson, T. E., Lazowska, E. D., & Levy, H. M. 1991, ACM Trans. Comput. Systems, 9, 175
Bradski, G., & Kaehler, A. 2008, Learning OpenCV: Computer Vision with the OpenCV Library (Sebastopol, CA: O'Reilly Media, Inc)
Chapman, B., Jost, G., & van der Pas, R. 2008, Using OpenMP: Portable Shared Memory Parallel Programming (Cambridge, MA: MIT Press)
Giles, M. K., Seward, A. J., Vorontsov, M. A., Rha, J., & Jimenez, R. 2000, Proc. SPIE, 4124, 89
Gropp, W., Lusk, E., & Skjellum, A. 2014, Using MPI: Portable Parallel Programming with the Message-Passing Interface (Cambridge, MA: MIT Press)
Hu, L., Xuan, L., Cao, Z., Mu, Q., Li, D., & Liu, Y. 2006, Opt. Exp., 14, 11911
Kong, L., Zhang, L., Zhu, L., Bao, H., Guo, Y., Rao, X., Zhong, L., & Rao, C. 2016, Chin. Opt. Lett. 14, 100102

Kwon, K. S., & Ready, S. 2015, Practical Guide to Machine Vision Software: An Introduction with LabVIEW (Weinheim: Wiley-VCH)
Langlois, M., Moretto, G., Richards, K., Hegwer, S., & Rimmele, T. R. 2004, Proc. SPIE, 5490, 59
Lee, J. H., et al. 2006, J. Korean Phys. Soc., 49, 139
Lee, J. H., et al. 2017, Curr. Opt. Photonics, 1, 107
Löfdahl, M. G. 2010, A&A, 524, A90
Michau, V., Rousset, G., & Fontanella, J. 1993, in Real Time and Post Facto Solar Image Correction, Proc. 13th National Solar Observatory/Sacramento Peak Summer Workshop, ed. R. Radick (Sacramento Peak: National Solar Observatory), 124
Rao, C., et al. 2010, Chin. Opt. Lett., 8, 966
Ren, D., & Dong, B. 2012, Opt. Eng., 51, 1705
Ren, D., Dong, B., Zhu, Y., & Christian, D. J. 2012, PASP, 124, 247
Ren, D., Jolissaint, L., Zhang, X., Dou, J., Chen, R., Zhao, G., & Zhu, Y. 2015, PASP, 127, 469
Ren, D., Penn, M., Plymate, C., Wang, H., Zhang, X., Dong, B., Brown, N., & Denio, A. 2010, Proc. SPIE, 7736, 77363P
Ren, D., & Zhao, G. 2016, PASP, 128, 105002
Rimmele, T. R. 2004, Proc. SPIE, 5490, 34
Rimmele, T. R., et al. 2004, Proc. SPIE, 5171, 179
Schmidt, D., Berkefeld, T., Heidecke, F., Fischer, A., von der Lühe, O., & Soltau, D. 2014, Proc. SPIE, 9148, 91481T
Smith, L., & Bull, M. 2001, Sci. Programming, 9, 83
Taylor, T. S., & Gregory, D. A. 2002, Optics Laser Tech., 34, 665
Thomas, S. 2004, Proc. SPIE, 5490, 766
Tyson, R. K. 2010, Principles of Adaptive Optics, 3rd ed. (Boston: Academic Press)
Tyson, R. K. 2015, Principles of Adaptive Optics, 4th ed. (Boca Raton, FL: CRC press)
van der Luehe, O., Soltau, D., Berkefeld, T., & Schelenz, T. 2003, Proc. SPIE, 4853, 187