Real-time liquid-crystal atmosphere turbulence simulator with graphic processing unit

Lifa Hu,[*] Li Xuan, Dayu Li, Zhaoliang Cao, Quanquan Mu, Yonggang Liu, Zenghui Peng, and Xinhai Lu

Changchun Institute of Optics, Fine Mechanics and Physics, Dongnanhu Street 16, Changchun, Jilin 130033, China
[*]Corresponding author: hulifa@ciomp.ac.cn

Abstract: To generate time-evolving atmosphere turbulence in real time, a phase-generating method for our liquid-crystal (LC) atmosphere turbulence simulator (ATS) is derived based on the Fourier series (FS) method. A real matrix expression for generating turbulence phases is given and calculated with a graphic processing unit (GPU), the GeForce 8800 Ultra. A liquid crystal on silicon (LCOS) with 256×256 pixels is used as the turbulence simulator. The total time to generate a turbulence phase is about 7.8 ms for calculation and readout with the GPU. A parallel processing method of calculating and sending a picture to the LCOS is used to improve the simulating speed of our LC ATS. Therefore, the real-time turbulence phase-generation frequency of our LC ATS is up to 128 Hz. To our knowledge, it is the highest speed used to generate a turbulence phase in real time.

OCIS codes: (230.6120) Spatial light modulators; (010.1330) Atmospheric turbulence; (160.3710) Liquid crystals; (010.1080) Adaptive optics.

References and links

1. M. A. van Dam, D. Le Mignant, and B. A. Macintosh, "Performance of the Keck Observatory adaptive-optics system," Appl. Opt. **43**, 5458–5467 (2004), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-43-29-5458.
2. M. Schöck, D. Le Mignant, G. A. Chanan, P. L. Wizinowich, and M. A. van Dam, "Atmospheric turbulence characterization with the Keck adaptive-optics systems. I. Open-loop data," Appl. Opt. **42**, 3705–3720 (2003), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-42-19-3705.
3. T. Aruga and Y. Kohyama, "Recovery of blurred images obtained through turbulent media," Appl. Opt. **42**, 190–203 (2003), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-42-2-190.
4. P. Hickson, "Impact of telescope seeing on laser adaptive optics," Opt. Lett. **31**, 2127–2129 (2006), http://www.opticsinfobase.org/ol/abstract.cfm?URI=ol-31-14-2127.
5. T. S. Taylor, A. S. Kransteuber, D. A. Gregory, and J. L. McClain, "Optical processing through turbulent media," SPIE **2752**, 292–298 (1996).
6. O. Keskin, L. Jolissaint, and C. Bradley, "Hot-air optical turbulence generator for the testing of adaptive optics systems: principles and characterization," Appl. Opt. **45**, 4888–4897 (2006), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-45-20-4888.
7. T. Kelly, D. F. Buscher, P. Clark, C. Dunlop, G. Love, R. M. Myers, R. Sharples, and A. Zadrozny, "Dual-conjugate wavefront generation for adaptive optics," Opt. Express **7**, 368–374 (2000), http://www.opticsinfobase.org/oe/abstract.cfm?URI=oe-7-11-368.
8. D. J. Butler, S. Hippler, S. Egner, W. Xu, and J. Bähr, "Broadband, static wave-front generation: Na-Ag ion-exchange phase screens and telescope emulation," Appl. Opt. **43**, 2813–2823 (2004), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-43-14-2813.
9. S. Hippler, F. Hormuth, D. J. Butler, W. Brandner, and T. Henning, "Atmosphere-like turbulence generation with surface-etched phase screens," Opt. Express **14**, 10139–10148 (2006), http://www.opticsinfobase.org/oe/abstract.cfm?URI=oe-14-22-10139.
10. B. M. Welsh, "Fourier-series-based atmospheric phase screen generator for simulating anisoplanatic geometries and temporal evolution," Proc. SPIE **3125**, 327–338 (1997).
11. J. Kolb, E. Marchetti, S. Tisserand, F. Franza, B. Delabre, F. Gonté, R. Brast, S. Jacob, and F. Reversat, "MAPS: a turbulence simulator for MCAO," Proc. SPIE **5490**,794–804 (2004).
12. J. S. Tharp and R. K. Tyson, "Measurement of the optical path difference over an atmospheric turbulence phase plate," Proc. SPIE **5490**, 805–809 (2004).

#105326 - $15.00 USD Received 16 Dec 2008; revised 23 Feb 2009; accepted 4 Mar 2009; published 17 Apr 2009

13. J. J. Widiker and E. P. Magee, "Open-loop simulations of atmospheric turbulence using the AdAPS interface," Proc. SPIE **5894**, 589404-1–589404-10 (2005).
14. J. D. Phillips, M. E. Goda, and J. Schmidt, "Atmospheric turbulence simulation using liquid-crystal spatial light modulators," Proc. SPIE **5894**, 589406-1–589406-11 (2005).
15. M. R. Brooks and M. E. Goda, "Atmospheric simulation using a liquid-crystal wavefront controlling device," Proc. **SPIE** 5553, 258–268 (2004).
16. T. S. Taylor and D. A. Gregory, "Laboratory simulation of atmospheric turbulence-induced optical wavefront distortion," Opt. Laser Technol. **34**, 665–669 (2002).
17. T. Kelly, G. D. Love, D. F. Buscher, R. M. Myers,C. N. Dunlop, Andrew Zadrozny and Ray M. Sharples, "Dual-conjugate wavefront generation with liquid-crystal spatial light modulators," Proc. SPIE **3749**, 662–663 (1999).
18. L. Hu, L. Xuan, Z. Cao, Q. Mu, D. Li, and Y. Liu, "A liquid-crystal atmospheric turbulence simulator," Opt. Express **14**, 11911–11918 (2006), http://www.opticsinfobase.org/oe/abstract.cfm?URI=oe-14-25-11911.
19. D. Dayton, S. Sandven, S. Browne, and J. Gonglewski, "Multi-segment spatial light modulators for the simulation of Kolmogorov turbulence," Proc. SPIE **3432**, 73–84 (1998).
20. F. Assémat, R. Wilson, and E. Gendron, "Method for simulating infinitely long and non-stationary phase screens with optimized memory storage," Opt. Express **14**, 988–999 (2006), http://www.opticsinfobase.org/oe/abstract.cfm?URI=oe-14-3-988.
21. J. Lipowski, "Real-time data processing on graphics processors," Proc. SPIE **6159**, 61594M-1–61594M-6 (2006).
22. T. Schiwietz, S. Bose, J. Maltz, and R. Westermann, "A fast and high-quality cone beam reconstruction pipeline using the GPU," Proc. SPIE **6510**, 65105H-1–65105H-12 (2007).
23. R. Frehlich, "Simulation of laser propagation in a turbulent atmosphere," Appl. Opt. **39**, 393–397 (2000), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-39-3-393.
24. J. D. Schmidt, M. E. Goda, and B. D. Duncan, "Aberration production using a high-resolution liquid-crystal spatial light modulator," Appl. Opt. **46**, 2423–2433 (2007), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-46-13-2423.
25. L. Hu, L. Xuan, Y. Liu, Z. Cao, D. Li, and Q. Mu, "Phase-only liquid-crystal spatial light modulator for wavefront correction with high precision," Opt. Express **12**, 6403–6409 (2004), http://www.opticsinfobase.org/oe/abstract.cfm?URI=oe-12-26-6403.
26. Q. Mu, Z. Cao, D. Li, L. Hu, and L. Xuan, "Open-loop correction of horizontal turbulence: system design and result," Appl. Opt. **47**, 4297–4301 (2008), http://www.opticsinfobase.org/ao/abstract.cfm?URI=ao-47-23-4297.

1. Introduction

An adaptive optics (AO) system is necessary for large aperture telescopes to obtain clear images of space objects [1–5]. An AO system with a deformable mirror (DM), a microelectronic mechanical system (MEMS), and a liquid-crystal (LC) wavefront corrector have been investigated and demonstrated. However, during the design and optimization of these AO systems, it is valuable to evaluate and characterize their performances with an atmospheric turbulence simulator (ATS). Many kinds of ATSs are available [6–12]. One of the most important characteristics of an ATS is its speed, which corresponds to its ability to generate atmospheric turbulence with different Greenwood frequencies. An LC ATS has been investigated previously with promising results because of its many merits such as its high density of pixels, low cost, programmability, good repeatability, low power consumption, light weight, and so on [13–20]. However, an LC device has a large number of pixels and a slow rate of response. Generally, to simulate the turbulence in real time, one has to calculate gray map pictures according to a specific turbulence simulation method before experimentally simulating the turbulence. Then, after the pictures are loaded into the buffer of the LC ATS electric board, they are displayed during the experiment to generate aberration wavefronts by modulating the incident light. It therefore avoids the problem of a long calculation time during the experiments. However, it is not a real-time simulation, and it is a very inconvenient application for testing and characterizing optical systems when one needs to change the turbulence simulating parameters. Therefore, although the LC ATS is programmable and free of mechanical movement, it has not been widely used.

In this paper we report a real-time calculation method for generating turbulence wavefronts and demonstrate it on our LC ATS. In Section 2, to solve the computation time problem we present an improved turbulence simulation method and use a graphic processing unit (GPU) to calculate the atmospheric turbulence phase in real time. Our method is based on the Fourier series (FS) [10,14,15] and is programmed with the Compute Unified Device Architecture (CUDA) to calculate the turbulence phase in real time. In addition, a parallel controlling method is also used in our LC ATS program that improves the simulating speed. In Section 3, experimental results of simulated turbulence for different coherence lengths are demonstrated and discussed. Finally, conclusions are given in Section 4.

2. Simulation algorithm

The modal method based on the FS expansion of the wavefront phase over a square area of dimension D_p is very successful. The major drawback is the long computation time involved in generating each screen, but the main advantage of the FS method is its vector–matrix representation, which allows it to be calculated with a GPU. Therefore, to overcome the computation time problem, we could modify the simulation equations and calculate the turbulence phase with a GPU. However, there are only limited functions available for mathematic calculation in the software development kit (SDK 1.0) of CUDA supplied by Nvidia Corp. [21,22]. Also, it cannot do some special calculations, such as complex matrix multiplication, and therefore it is necessary to modify the expressions and remove any complex calculation.

First, we summarize the physical idea for turbulence simulation and the main procedure of the FS method, and its detailed derivation procedure can be found in [10]. The turbulence phase could be expressed by the FS with finite terms. The coefficients of every term of the FS could be generated randomly. It should be noted that these coefficients are independent random variables and have a mean square value determined by the turbulence power spectrum density (PSD). First, the Von Karmen turbulence phase PSD is also used. It has the following form:

$$\Phi(f_r) = \frac{0.00969k^2 \int \mathrm{d}z C_n^2(z)}{\left(|f_r|^2 + L_0^{-2}\right)^{11/6}}, \tag{1}$$

where f_r is the spatial frequency, C_n^2 is the refractive structure constant, L_0 is the outer size of the turbulence, and integration is done along the altitude. An FS with finite terms is used to represent the random wavefront phase [10]:

$$\phi_k(\vec{r}) = 2\cdot\mathrm{Re}\left[\sum_{n=0}^{N-1}\sum_{n'=0}^{N-1} c_{n,n'}\exp\left\{j2\pi\left(\frac{n\hat{x}}{D_p}+\frac{n'\hat{y}}{D_p}\right)\right\} + \sum_{n=1}^{N-1}\sum_{n'=-(N-1)}^{-1} c_{n,n'}\exp\left\{j2\pi\left(\frac{n\hat{x}}{D_p}+\frac{n'\hat{y}}{D_p}\right)\right\}\right], \tag{2}$$

where $(2N-1)^2$ is the number of terms included in the FS representation of ϕ_k and $c_{n,n'}$ is the FS coefficient. The quantities $\hat{x}$ and $\hat{y}$ are unit vectors along the x and y directions, respectively. $\vec{r} = \hat{x}\cdot m\cdot\frac{D_p}{N} + \hat{y}\cdot m'\cdot\frac{D_p}{N}$, where m and m' are integers from 0 to N. It should be noted that we also used a uniformly spaced frequency grid for generating the FS coefficients as used in [10,13,14,23]. However, this creates phase screens that under-represent the low spatial frequency fluctuations. To avoid the problem, we used 256 coefficients of the FS to properly represent low-frequency fluctuations. In fact, the GPU has a high capability of parallel calculation. Therefore, the large number of coefficients will not be a calculation problem. Although a more accurate method is to use a nonuniform grid with more samples allocated to the low spatial frequencies, it will make it difficult for a matrix calculation to generate an atmospheric turbulence phase. It should be noted that the imaginary part of the expression inside the square brackets of Eq. (2) is thrown away. The expression in Eq. (2) can easily be expressed in a matrix algebra notation:

$$\phi_k(\vec{r}) = 2\cdot\mathrm{Re}\left\{\left(\exp(j2\pi f_x)\right)^T\cdot\left[C_L\cdot\exp(j2\pi f_x) + C_R\cdot\exp(-j2\pi f_y)\right]\right\}, \tag{3}$$

#105326 - $15.00 USD Received 16 Dec 2008; revised 23 Feb 2009; accepted 4 Mar 2009; published 17 Apr 2009

where C_L and C_R are the matrices of dimension N×N containing the complex random variables $c_{n,n'}$, which correspond to the left and right terms in Eq. (2). f_x and f_y are spatial frequency matrixes. However, it will take a long time to generate the turbulence phase. In addition, it cannot be calculated with a GPU. Therefore, we have to improve the method for a real-time calculation of the atmospheric turbulence phase.

Next, to calculate the turbulence phase with a GPU, we have to modify Eq. (2). Therefore, it is necessary to yield the matrix algebra notation without a complex number from Eq. (2). Removing the imaginary part of Eq. (3), we get

$$\phi(\vec{r}) = 2\left(x_c^T C_1 y_c + x_s^T C_2 y_s + x_c^T C_3 y_s - x_s^T C_4 y_c\right), \tag{4}$$

where, $x_c = \cos\left(2\pi f_x\right)$, $x_s = \sin\left(2\pi f_x\right)$, $y_c = \cos\left(2\pi f_y\right)$, $y_s = \sin\left(2\pi f_y\right)$, and f_x and f_y depend on the wind speed v and the time during the real-time turbulence simulation. v is equal to $2.39 f_g r_0$, where f_g is the Greenwood frequency and r_0 is the atmosphere coherence length. In addition, C_1, C_2, C_3, and C_4 are the real parts of (C_R + C_L), (C_R - C_L), (C_R - C_L), and (C_R + C_L), respectively. In fact, they could be calculated during initializing a turbulence phase simulation program in real time. Also, C_R and C_L are related to the PSD of the atmosphere turbulence:

$$C_L = \sqrt{\Phi(\vec{f})/D_p^2} \cdot (r_1 + i \cdot r_2), \tag{5}$$

$$C_R = \sqrt{\Phi(\vec{f})/D_p^2} \cdot (r_3 + i \cdot r_4), \tag{6}$$

where, r_1, r_2, r_3, and r_4 are two independent random draws from a Gaussian probability density function because the FS coefficients obey circular complex Gaussian statistics. The first column and row of C_L and C_R should be zero. According to Eq. (4), it is easy to generate the turbulence phase.

Now we can use CUDA as a programming environment for calculation with a GPU according to the above theory [10]. A GPU with GeForce 8800 Ultra from Nvidia Co. was used in our experiment. Its technical specifications include a GPU clock of 1.5 GHz, a memory clock of 1.08 GHz, a memory of 768 Mbytes, a memory bandwidth of 103.7 GB/s, 128 stream processors, and a calculation capability of 576 GFLOPs (floating-point number operations per second). In fact, CUDA is developed for general-purpose computing, and this makes a GPU as a computing device capable of executing a very high number of threads in parallel. A GPU has several multiprocessors capable of simultaneously executing parallel threads of the same program on different data elements. When a multiprocessor program referred to as kernel is executed on a multiprocessor, a grid composed of thread blocks is created to run the same kernel on more than one multiprocessor. The number of blocks is equal to 256 in our simulation. In turbulence phase generation, the block size is defined as 16. In addition, each thread in a block is responsible for computing the gray level value of one pixel of a LC on silicon (LCOS). It should be noted that the vectors of x_c, x_s, y_c, and y_s in Eq. (3) are transformed to corresponding matrices with non-zero values in the first column. The turbulence gray map calculations with CUDA in a GPU are mainly based on matrix multiplication and addition. It should be noted that the cos and sin functions will take 32 clock cycles, which will dominate the calculation time during real-time turbulence gray map generations.

3. Experiment and discussion

The experiment setup including the LC ATS is shown in Fig. 1. The LC molecular alignment can be realigned with applied voltages, which would change its effective refractive index [24,25]. Thus, the phase of incident light can be modulated by an LC device. In our experiment, the LC device is a LCOS that is set as a reflective surface for the original wave. Our LC ATS includes a LCOS with 256×256 pixels (from Boulder Nonlinear Systems Co.) and a host computer with a CPU Intel® Core 2 2.66 GHz, 1 GB DDR2, and a GPU

GeForce8800 Ultra. The operating system is Windows® XP. A 633 nm laser output from a fiber-bound travels through the collimating lens L1 with a focus length of 30 mm and a spatial filter (SF) setup with an aperture size of 15 µm (from Newport Co.), which is used to generate an ideal spherical wavefront. Then the laser travels through another collimating lens L2 with a focus length of 100 mm and a beam splitter (BS1). The BS1 sends the aberrated beam modulated by the LCOS 256 to our LC AO system [26]. The aberrated beam passes through the first relay lenses L3 and L4. Then it is reflected by the folding mirrors M2 and M3. The *p* polarization component of the aberrated beam is reflected through the second relay lenses L5 and L6 by the polarized beam splitter (PBS). It can be modulated by our LCOS because its polarization direction is parallel to the LC molecular alignment direction. The *p* polarization light passing through the third relay lenses L7 and L8 goes to a LCOS 512 for background aberration correction. The incident angle on the LCOS 512 is about 5° so that the reflected beam can be separated from the incident light by M5. The reflected light then goes to another beam splitter (BS2). The L9 lens is used to collimate the beam. One part of the light goes to a Shack–Hartmann wavefront sensor (SH WFS) for phase reconstruction through BS2, the other to a CCD for intensity measurement. The SH WFS ShaH 1000 from Moscow State University is used to measure the wavefront with a frequency of 500 Hz. Its effective microlens number is 415 with hexagonal arrangement. The SH WFS's CCD is the Andor DU860 CCD with a pixel number of 128×128, and a pixel size of 24 µm. The CCD for imaging is the Andor DU897 with a pixel number of 512×512 and a pixel size of 15 µm. It should be noted that the response time of the LCOS 256 is about 2.6 ms when the LCOS is at 35°C. Both the LC refractive index and the response speed depend on the temperature. Increasing the temperature of the LC layer will improve its response speed. However, it will also decrease its effective refractive index. Therefore, a heater is used on the LCOS so that 35°C is set as the working temperature of the LC layer in the LCOS. Because of its high response speed and relatively small pixel numbers, we used it as a real-time turbulence simulator. The LCOS 512 is used as a compensator to correct the static aberration in the optical layout.

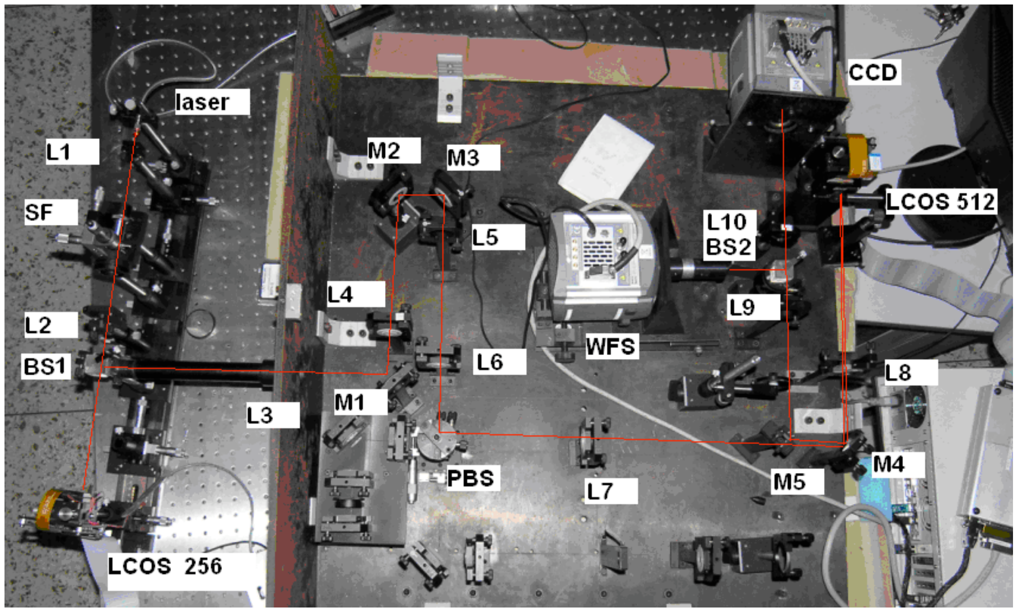


Fig. 1. Optical layout to characterize the LC atmospheric turbulence simulator.

In addition to turbulence and gray calculation, two other factors affecting the speed of our LC ATS are also considered. First, the relationship between the phase retardation and the applied voltage of the LCOS is not linear, as shown in Fig. 2. Therefore, a look-up table

(LUT) for gamma correction is needed to realize the linear characteristics as a calibration. The equation of the fitted curve in Fig. 2 is for the LUT. Gamma correction is a time-consuming process if one uses a CPU or digital signal processing (DSP) to do it because of the large number of pixels and complex computation equation. To save time, the LUT is also conducted in the GPU with the CUDA kernel function.

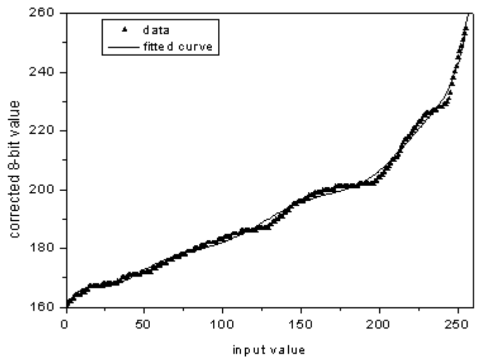


Fig. 2. Gamma correction curve for the LCOS.

Second, a multithread technique is also used; that is, after gamma correction with the LUT, a gray map is sent to the LCOS in one thread and at the same time another picture is generated in the other thread, as shown in Fig. 3. Specifically, in the control program of our LC ATS, calculation and sending threads were used: one was to generate a turbulence gray map as noted by thread 1 in Fig. 3, and the other was to send the gray map to the LCOS and wait for the LC response, as noted by thread 2 in Fig. 3. It should be noted that the time to change the parameters of time and corresponding coordination can be neglected. We recorded the time for both threads when the LC ATS was simulating. The results of their times are shown in Fig. 4. Thread 1 took an average of 7.8 ms, as denoted by the star symbol in Fig. 4. In addition, the LC response time was about 2.6 ms with the heater set at 35°C, and the data transmission time was about 3.4 ms. The time of 3.4ms for a frame is an average value out of sending 100 pictures to LCOS. Therefore, the time for the second thread was on average about 6 ms, as denoted by the cross in Fig. 3. Before sending another turbulence gray map to the LC ATS, thread 2 will wait until the GPU calculation of thread 1 has ended. Because of our parallel multithread algorithm, a longer time thread determined the speed of our LC ATS. That is, the effective time delay of our LC ATS was 7.8 ms, and the speed of our LC ATS with the multithread control method was about 128 Hz. To our knowledge, it is the fastest real-time nematic LC ATS. It should be noted that the rapid development of GPU hardware is promising for decreasing the calculation time to less than 2.6 ms, and that was the response time of our LCOS. In addition, after the peripheral component interconnect (PCI) electronic board of the LCOS is replaced by the PCI Express (PCI-E) with a much higher bandwidth of data transmission, the time for sending a gray map theoretically will be less than 1 ms. As a comparison, the algorithm to calculate gray maps has also been implemented in the CPU using the same computer with Matlab, and the total time to generate 100 gray maps with our Matlab program was obtained.

Then we found that the average time to generate one gray map was 217 ms. In addition, we also completed a C version calculation program according to our improved method, and the average time was about 26.9 ms to generate a turbulence phase. It is still much larger than that of a GPU. In fact, the GFLOPS performance of the GF 8800 Ultra is 576 GFLOPS, and it is not larger than 51 GFLOPS for a Intel® Core 2 CPU. In addition, the calculation capability of a CPU depends on its clock frequency and that is more and more difficult to improve. However, the calculation capability of a GPU depends on both its frequency and number of stream processors. Therefore, it is an easy and effective way to improve its parallel calculation capability by increasing the number of stream processors. In the future, the GFLOPS performance of a GPU will be higher and higher than that of a CPU. The advantages offered by GPUs when processing relatively large data sets will be more and more significant.

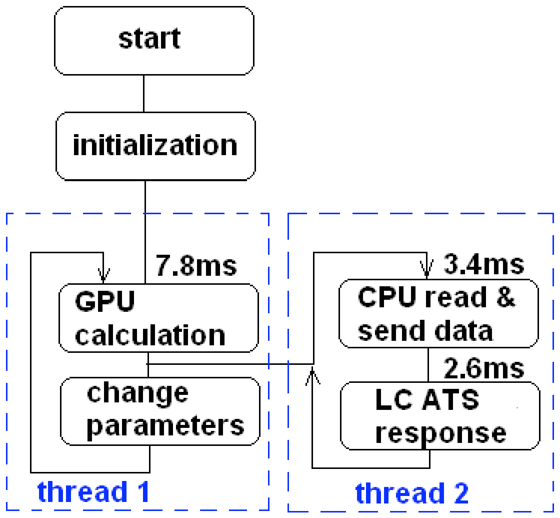


Fig. 3. Multithread control of the LC atmospheric turbulence simulator (LC ATS).

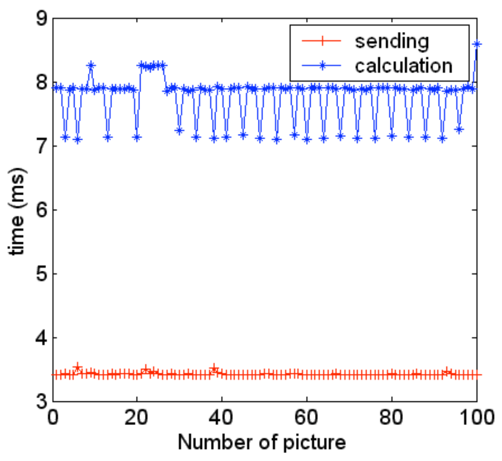
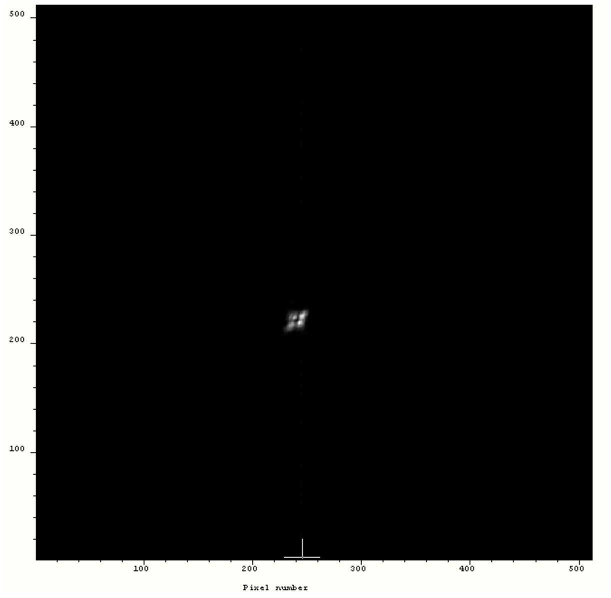


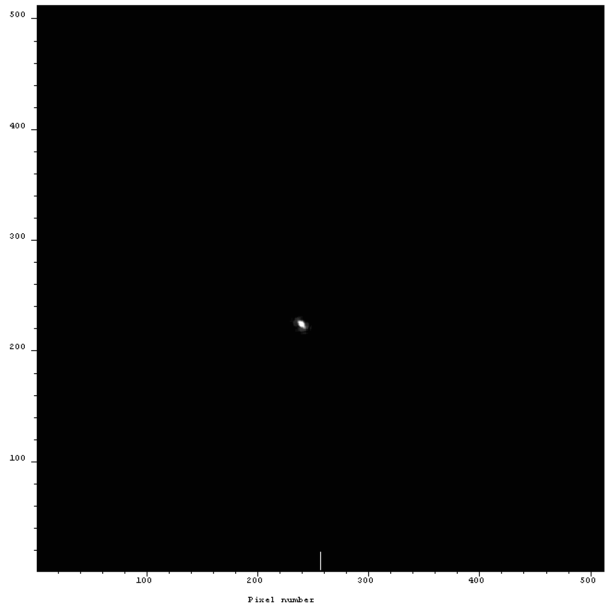
Fig. 4. Time for every picture in our LC ATS. Crosses denote the total time of sending a picture and waiting for the LC response; stars denote the total time of generating wavefronts and gamma correction in the GPU.

The phase structure function is often used to characterize a turbulence simulation method. In fact, we measured the simulated turbulence with a wavefront sensor and calculated its phase structure function. Although the phase structure function as a function of distance has a similar tendency as that described by Eq. (21) in [10] and has a similar power index, they have different amplitudes. Continued work has to be done to resolve the problem. The next is to replicate the calculated turbulence wavefront with our LC ATS with high precision, which is not a problem because of the high density of LC pixels.

Before simulation, the aberration of the optical layout as shown in Fig. 1 is compensated statically by the LCOS 512. After correction, the peak-to-vale (PV) and rms of the aberration is 1.08 λ and 0.19λ, respectively, and the image acquired by the CCD is shown in Fig. 5. The unit of x and y axis is in pixels, and the CCD has 512×512 pixels. The main aberration is the defocus, as shown in Fig. 5(a); after correction, the image is clear and the defocus is removed, as shown in Fig. 5(b).



(a)



(b)

Fig. 5. Image measured with DU 897 CCD in optical layout; the unit along *x* and *y* is in pixels (a) before correction, (b) after correction.

A series of turbulence was simulated with our LC ATS, and the main parameters are defined as follows: the simulation aperture of the telescope is 1 m, the turbulence outer size is 50 m, the zenith angle is zero, the velocity along *x* and *y* are assumed to be the same for simplification with values ranging from 0.5 m/s to 21 m/s, and C_n^2 is from 5×10^{-17} to 5×10^{-16} as shown in Table 1. The three wind velocities were chosen to get low and high Greenwood frequencies. As a result, the corresponding r_0 obtained ranged from 4.01 cm to 14.07 cm, and f_g ranged from 2.46 Hz to 8.53 Hz as shown in Table 1. Figure 6 shows the corresponding videos of the light source disturbed by the LC ATS. It should be noted that the area showing the image is part of the CCD, that is, from the 200th pixel to the 260th pixel along x and y, respectively. Generally, a larger r_0 leads to larger aberrations. It is noticeable that the smaller r_0 leads to a more blurred image, which can be interpreted with the rms results of the wavefront, as shown in Fig. 7. A series of wavefronts acquired with the WFS, as shown in Fig. 7, were obtained while the LC ATS was running for different parameter settings. It can be seen that the smallest r_0 of 4.01 cm has the largest variations of rms.

Table 1. Parameters of atmospheric turbulence simulation

$C_n^2(\times10^{-17})$	r_0 (cm)	f_g (Hz) (V_x=V_y = 0.5m/s)
6.3	14.07	2.46
8.2	12.01	2.88
10	10.66	3.249
16	8.04	4.3
26	6.01	5.76
50	4.01	8.53

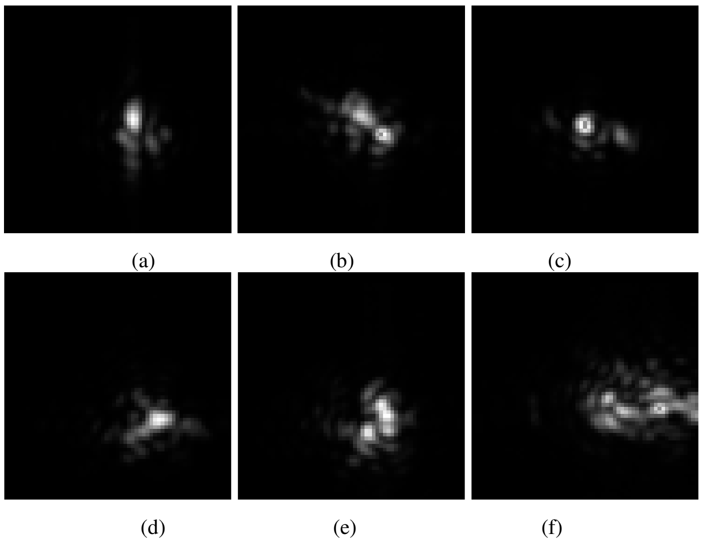


Fig. 6. Image of the light source disturbed by the LC ATS with different r_0. The area is from the 200th to 260th pixel from the CCD along x and y, respectively. (a) r_0 = 14.07 cm (Media 1). (b) r_0 = 12.01 cm (Media 2). (c) r_0 = 10.66 cm (Media 3). (d) r_0 = 8.04 cm (Media 4). (e) r_0 = 6.01 cm (Media 5). (f) r_0 = 4.01 cm (Media 6). (736K per file)

#105326 - $15.00 USD Received 16 Dec 2008; revised 23 Feb 2009; accepted 4 Mar 2009; published 17 Apr 2009

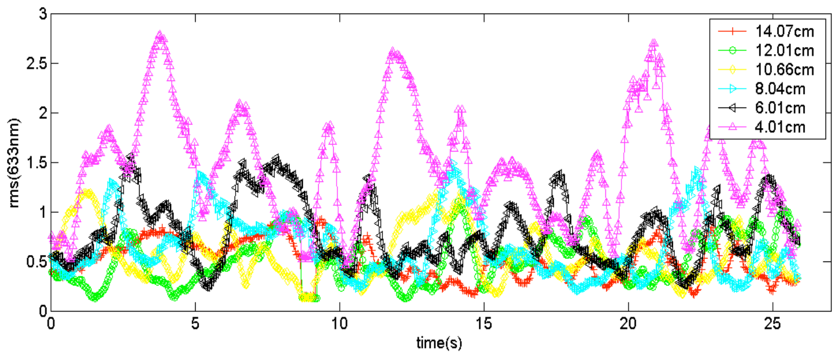


Fig. 7. Image of the light source disturbed by the LC ATS with different r_0. (a) $r_0 = 14.07$ cm. (b) $r_0 = 12.01$ cm. (c) $r_0 = 10.66$ cm. (d) $r_0 = 8.04$ cm. (e) $r_0 = 6.01$ cm. (f) $r_0 = 4.01$ cm.

Although the LC ATS has many merits as mentioned in Section 1, its first disadvantage is that its linear response is only in limited phase dynamics. LC ATS generates a turbulence phase through binary optics diffraction, and the precision of a generated wavefront depends on the number of pixels. If the PV of a simulated turbulence wavefront is so large that the number of pixels is not enough to keep a high diffractive efficiency of the LCOS, the error of the generated wavefront will be large. However, a small r_0 often means that the PV of atmospheric turbulence wavefront will be large. Therefore, a LC ATS with a finite number of LC pixels can not simulate an atmospheric turbulence with a relatively small r_0.

4. Conclusion

A prototype LC ATS for testing our AO systems has been studied. A turbulence generation method based on real matrix calculation was verified theoretically. The generation of a turbulence gray map to drive the LCOS is based on a GeForce 8800 Ultra GPU. Our LC ATS's speed is up to 128 Hz in real-time simulation. With the calculation speed improvement of the GPU hardware, the speed of our LC ATS could be increased greatly. Our experimental results demonstrate that an LC ATS is feasible for use in real time when simulating turbulence in laboratory conditions.

Acknowledgments

This work is supported by the National Natural Science Foundation grants 60736042, 60578035, 60811120025, and 50703039. The authors thank all of the anonymous reviewers for their reasonable suggestions.