Adaptive cross-correlation algorithm for extended scene Shack–Hartmann wavefront sensing

Erkin Sidick,* Joseph J. Green, Rhonda M. Morgan, Catherine M. Ohara, and David C. Redding

Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, USA

**Corresponding author: Erkin.Sidick@jpl.nasa.gov*

Received October 5, 2007; revised November 20, 2007; accepted December 16, 2007; posted December 21, 2007 (Doc. ID 88319); published January 16, 2008

We present an adaptive cross-correlation algorithm for a large dynamic range extended-scene Shack–Hartmann wavefront sensor. We show that it accurately measures very fine image shifts over many pixels under a variety of practical imaging conditions. © 2008 Optical Society of America

OCIS codes: 100.5070, 120.5050, 010.7350, 010.1080, 110.6770.

The Shack–Hartmann sensor (SHS) is an optical instrument widely used for wavefront sensing in optical testing and astronomical adaptive optics. In its simplest form, an SHS consists of a lenslet array and a detector. The conventional SHS uses a point source such as a star, a laser, or a pinhole as its object. In such a case, the image captured by a Shack–Hartmann (SH) camera is an array of spots, each of which is a point spread function of the associated lenslet. The wavefront error incident on the lenslet array produces position shifts of the spots, and the overall wavefront error is determined from the values of those shifts. The spot locations are usually determined by calculating the centroid (center of mass) of the spot images [1].

In some applications, such as solar telescopes [2], remote imaging along short horizontal or slant paths from the ground [3], and remote imaging from space, a point source is not available, but an extended scene is. In such a case, the lenslet array forms an array of subimages or cells at the detector. The incident wavefront error causes the subimages to shift; therefore, the wavefront error can be determined by estimating the shifts of all the subimages from their original positions. This is essentially a problem of image registration or aligning two images with each other, which is solved using cross-correlation [4] when the allowable transformations include a small range of rigid transformations (translation, rotation, and scale changes). In this approach, one typically computes the cross-correlation (CC) between target and reference images and determines the values of transformation parameters from the location of the CC peak. Indeed, Rimmele and Radick used a correlation technique referred to as "correlating Shack–Hartmann" for wavefront sensing using images of solar granulation in an extended-scene SHS (ES-SHS) [2]. This technique was further investigated by other researchers for arbitrary scene content having different illumination and noise levels [3,5,6]. Recently, Knutsson and Petersen proposed a new approach that estimates the shift based on the phase of two images' cross-correlation spectrum [7]. The studies mentioned above are restricted to situations where each lenslet image is shifted but not distorted and the mutual shift between two images is less than one pixel. Shift estimation is limited by errors due to the wraparound of pixels in the localization of CC peak. Estimation accuracy is also reduced when the two images are distorted with respect to each other, because the peak location of the CC function becomes obscured. The technique proposed in [7] uses a single phase slope-fitting iteration and thus can suffer from low accuracy when the shift is large and/or the image is distorted, as in the case of slant path imaging where the lenslet images are distorted by anisoplanatism effects.

We have developed a new algorithm for ES-SHS that works for large dynamic range (shifts of up to several pixels) and for distorted images. It is also based on the phase of CC spectrum as proposed in [7], but it carries out the phase slope-fitting process in an iterative or adaptive manner and avoids pixel wraparound errors by using a larger subimage size for image shifting and a smaller subimage size for phase slope fitting. We call this algorithm an adaptive cross-correlation (ACC) algorithm. In this paper, we describe the ACC algorithm in detail first. Then we explain the trade-off between the computation cost and the shift estimate accuracy using measured data. In the end we compare the performance of the ACC algorithm when used with a point source and an extended scene.

Throughout the process of algorithm development, we worked with real image data measured in our ES-SHS testbed. The extended scene was a chrome bar target on a glass. Figure 1(a) is an example of the extended-scene image cells used to test the current algorithm. Details of this deformable-mirror (DM) based ES-SHS testbed and additional information on the experimental verification of this new ACC algorithm are presented elsewhere [8].

The ACC algorithm estimates the shift between two subimages or cells with the following steps:

(1) Choose an $N\times N$ pixel test cell, $S_i(x,y)$, from the center of a lenslet image, as shown with a white frame in Fig. 1(b), and compute its Fourier transform, $\hat{S}_i(u,v)$. Also, choose an $M\times M$ pixel reference cell, $r(x,y)$, and compute its Fourier transform, $\hat{r}(u,v)$, where $M<N$ and both are preferentially a power of 2 for this algorithm. The $r(x,y)$ should be

0146-9592/08/030213-3/$15.00

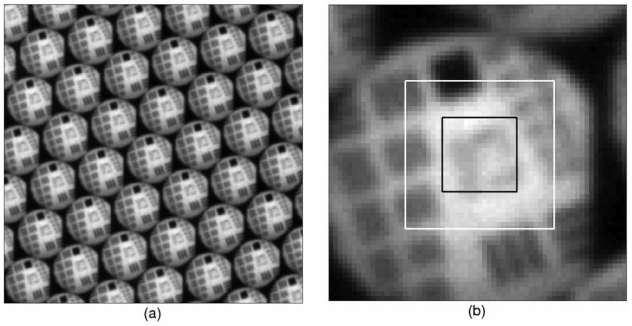

Fig. 1. (a) SH extended-scene subimage array obtained with a bar target on our SH wavefront sensor testbed. (b) One 64×64 pixel subimage of the SH extended scene. The white frame shows a 32×32 pixel ($N=32$) subaperture, and the black frame shows a 16×16 pixel ($M=16$) subaperture. These subapertures are referred to as big and small cells, respectively, in this paper.

chosen from the same frame as $S_i(x,y)$ but from a different lenslet image, preferably near the center of the whole image. An $M\times M$ cell is shown with a black frame in Fig. 1(b). Each lenslet image is approximately 65×65 pixels on our testbed. Therefore, we chose $N=32$ and $M=16$ to test our algorithm.

(2) Obtain a small test cell pair, $s_i(x,y)$ and $\hat{s}_i(u,v)$, where $s_i(x,y)$ is equal to the central $M\times M$ pixel portion of $S_i(x,y)$ and $\hat{s}_i(u,v)$ is its Fourier transform. Multiply $\hat{r}^*(u,v)$ and $\hat{s}_i(u,v)$ to obtain a CC function in the Fourier domain, $\hat{c}_i(u,v)=\hat{r}^*(u,v)\hat{s}_i(u,v)$, where * denotes a complex conjugate. After that obtain the phase function $\phi_i(u,v)$ of $\hat{c}_i(u,v)$.

(3) Fit u and v slopes, a and b, to $\phi_i(u,v)$ over a small $\{u,v\}$ subdomain using a standard least-square fit algorithm. The digitized version of $\phi_i(u,v)$, $\phi_i(n,m)$, is also 16×16 pixels in size, with $n=1,2,\ldots,16$ and $m=1,2,\ldots,16$, respectively. Its center is at $n=m=9$. We used eight frequency components of $\phi_i(n,m)$ for slope fitting, with $n=9,10,11$ and $m=9,10,11$, but excluding $n=m=9$. As pointed out by [7] and some references therein, the above choice has the advantages of being the least sensitive to aliasing and leaving the low-spatial-frequency, high-contrast components in the image but rejecting the high-spatial-frequency, low signal-to-noise ratio components completely from the shift estimate.

(4) Multiply $\hat{S}_i(u,v)$ by $\exp[-j2\pi(au+bv)]$, where a and b are phase slopes obtained in Step 3. After that compute the inverse fast Fourier transform (IFFT) of this product. This step is equivalent to shifting $S_i(x,y)$ so that its position aligns to that of the reference cell.

(5) Repeat Steps 2 through 4 in an iteration loop while accumulating a and b, until a maximum iteration number is reached or the change in image shift becomes smaller than a predetermined tolerance.

As we can see from Steps 2 and 4 above, a big cell is used for image shifting, and a small one for slope fitting. As mentioned before, this prevents the central $M\times M$ pixel portion of each cell from getting wrap-around error and also speeds up the calculation process. It should be noted that $\hat{r}(u,v)$ needs to be calculated only once for each SH image, and $\hat{S}_i(u,v)$ needs to be calculated only once for each test cell. As compared with other CC techniques, such as periodic correlation described in [5], the current approach is computationally much slower. This is because the main computational burden comes from 2D FFT and IFFT operations, and the ACC algorithm uses such operations more times than the periodic correlation. The exact difference in the computation time between these two algorithms depends on the total number of phase slope-fitting iterations used in the ACC algorithm.

There is a trade-off between the computation cost and the shift estimate accuracy of the ACC algorithm. Both of these factors also depend on the scene content, the image distortion level, the noise, and the illumination levels, and the amount of shift between a test cell and its reference. To understand such a trade-off of the ACC algorithm under a realistic condition, we took six frames of extended-scene signal images (Fig. 1) using different amounts of integration time. For the convenience of our discussion below, we will identify those frames with $t=t_l, l=1,2,\ldots,6$. The gray-level ranges of the corresponding small (16×16 pixel) test cells increased by a factor of more than 2 when the exposure time was increased from its minimum value ($t=t_1$) to the maximum ($t=t_6$). Also, in the last two cases of exposure time, some pixels of the small test cells got saturated. From each of those frames, we chose one reference cell, $r^{t_l}(x,y)$, and 10 test cells, $S_i^{t_l}(x,y)$ with $i=1,2,\ldots,10$, such that they are separated from each other by at least five lenslet images. The mean (in terms of t_l) intrinsic shifts of those 10 cells relative to the reference cell ranged from 0.20 to 0.82 pixels. Also, to obtain test cells with shifts larger than 1 pixel, we moved the white frame in Fig. 1(b) horizontally to the right to obtain $G_{ik}^{t_l}(x,y)=S_i^{t_l}(x-k\Delta x,y)$, where Δx is the width of each pixel and $k=0,1,2,3$. It should be emphasized that the $G_{ik}^{t_l}(x,y)$ cells obtained with different k values have not only different shifts but also slightly different structures due to the differences in the image textures of different lenslet images. In this way, we obtained 60 test cells for each k value (6 in t_l and 10 in i) to test the accuracy and speed of the ACC algorithm.

Figure 2(a) shows the shift estimate error as a function of actual shift obtained with three different values of shift-finding tolerance, and Fig. 2(b) shows the mean numbers of iterations required to achieve the estimate accuracy shown in Fig. 2(a). The error bars in Fig. 2(a) represent the standard deviation of the 60 estimate error values. The actual shift in Fig. 2 is defined as $(\Delta d_{\text{act}})_{ik}-(\Delta d_{\text{act}})_{i0}$ with $(\Delta d_{\text{act}})_{ik}=\sqrt{(\delta x_{i0}-k\Delta x)^2+\delta y_{i0}^2}$, and the estimate error is $(\Delta d_{\text{est}})_{ik}-(\Delta d_{\text{act}})_{ik}$ with $(\Delta d_{\text{est}})_{ik}=\sqrt{\delta x_{ik}^2+\delta y_{ik}^2}$, where $(\delta x_{ik},\delta y_{ik})$ is the shift estimate output of the ACC algorithm. As we can see, it takes only four iterations on the average to achieve 0.01 pixel estimate accuracy under the current experimental conditions. Although not shown here, we also obtained the estimate error and the mean number of iterations as a

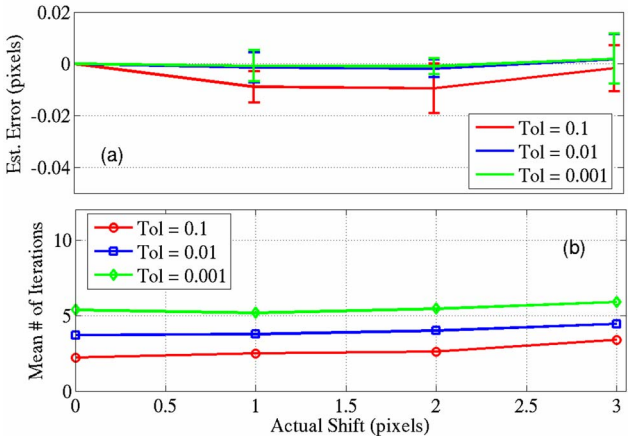

Fig. 2. (Color online) Comparison of (a) estimate errors and (b) average numbers of iterations corresponding to different image-shifting tolerances. The error bars in (a) correspond to the standard deviation of the estimate errors obtained from the 60 different images. See the main text for more information.

function of integration time for the case of tolerance $=0.01$ pixels. We had 40 data points for each t_l in this case (4 in k and 10 in i) and obtained peak-to-valley values of 0.004 pixels (mean error), 0.002 pixels (error standard deviation), and 1 (mean number of iterations). These results clearly show that (i) the estimate error is more or less stable whatever the actual shift is thanks to the iterative nature of the ACC, (ii) the number of iterations does not increase with the actual shift, and (iii) both the estimate error and the number of iterations are not sensitive to the signal level within the range considered in this paper. These are some of the most significant advantages of the ACC algorithm.

Figure 3 is an example of the wavefront estimates from the ACC algorithm when applied to a spot image and an extended-scene image, respectively. Here we poked the appropriate actuators on the DM to produce an "F" letter pattern first, then obtained the SH images of a point source and the extended scene, and after that processed the two images with the ACC algorithm to obtain the corresponding subimage offsets. The DM has 32×32 actuators with a 1 mm spacing, and the width of each camera pixel is 5 μm. Therefore, the actuator distance on the SH camera image is 200 pixels. Finally we reconstructed the optical path difference (OPD) maps corresponding to these two sets of subimage offsets using a Zernike-mode-based wavefront reconstructor. The left-hand portion of Fig. 3 is the result of the point source, and the right-hand portion corresponds to the extended scene. As we can see, the OPD maps of these two cases are almost identical. The difference near the right-hand edges of the OPD windows is probably caused by vignetting [8] of the subimages in those areas. To obtain a more quantitative comparison, we first calculated the histograms of the shift estimates, H_{spot} and H_{scene}, respectively, for the two cases with a bin of 0.02 pixels and for a range of 0–0.24 pixels. Then we calculated the percentage error of the two histograms, $E_{\mathrm{H}}=100\times(H_{spot}-H_{scene})/H_{spot}$. The obtained maximum and mean values of E_{H} are 9.1% and 3.0%, respectively. That is, the difference in the shift estimates is small.

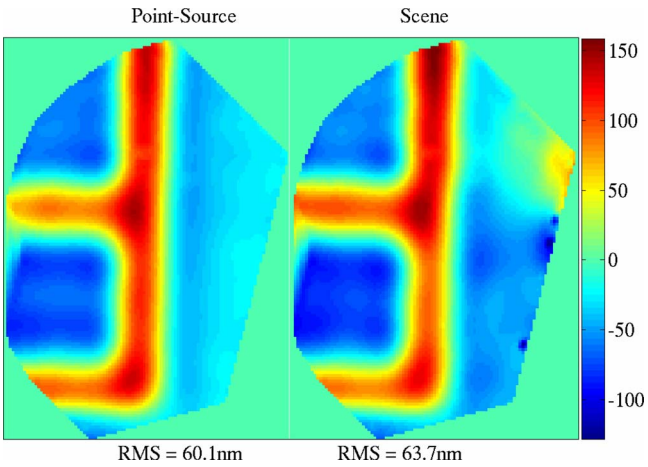

Fig. 3. (Color online) Comparison of OPD maps reconstructed from the offsets of subimages measured with a point source and a bar target extended scene, respectively. The units of the color-bar label are nanometers. The "F" letter pattern in each OPD map was produced by poking the appropriate actuators on the DM.

In conclusion, we have developed an ACC algorithm for a Shack–Hartmann wavefront sensor. It is based on the phase of the CC spectrum of two image cells. Working with measured image cells having multipixel image shifts and different signal levels, we have shown that the ACC algorithm can yield image shift estimates with high subpixel accuracy, such as 0.01 pixel, and its accuracy and computation time vary very little with the amount of actual shift up to more than 3 pixels as well as with the signal level. We expect that this algorithm will serve as an alternative to the existing image registration methods. We also expect it to become widely useful in both calibration and normal operation of large dynamic range SH wavefront sensing and control systems, including those used in ground- and space-based telescopes.

References

1. S. Thomas, T. Fusco, A. Tokovinin, M. Nicolle, V. Michau, and G. Rousset, Mon. Not. R. Astron. Soc. **371**, 323 (2006).
2. T. R. Rimmele and R. R. Radick, Proc. SPIE **3353**, 72 (1998).
3. L. A. Poyneer, D. W. Palmer, K. N. LaFortune, and B. Bauman, Proc. SPIE **5894**, 58940N-1 (2005).
4. L. G. Brown, ACM Comput. Surv. **24**, 325 (1992).
5. L. A. Poyneer, Appl. Opt. **42**, 5807 (2003).
6. V. Michau, J. M. Conan, T. Fusco, M. Nicolle, C. Robert, M. T. Velluet, and E. Piganeau, Proc. SPIE **6303**, 63030B (2006).
7. P. A. Knutsson and M. O. Petersen, Opt. Express **13**, 9527 (2005).
8. E. Sidick, R. M. Morgan, J. J. Green, C. M. Ohara, and D. C. Redding, Proc. SPIE **6687**, 668710 (2007).