

Direct demodulation of Hartmann–Shack patterns

Amos Talmi

Timi Technologies Ltd, Ramat Hashofet 19238, Israel

Erez N. Ribak

Department of Physics, Technion–Israel Institute of Technology, Haifa 32000, Israel

Received July 17, 2003; revised manuscript received October 27, 2003; accepted December 3, 2003

Hartmann–Shack wave-front sensors produce a distorted grid of spots whose deviation from perfection is linear with the wave-front gradient. Usually, the centroid of each spot is calculated to provide that deviation, but it is also possible to perform the calculation by Fourier demodulation of the spot pattern [Opt. Commun., 285, 2003]. We show that this demodulation can be performed directly on the grid, without reverting to Fourier transforms. Tracking the motion of each centroid individually is limited to well-defined spots with motions smaller than their pitch. In contrast, our method treats the image as a whole, is not limited to non-overlapping or sharp spots, and allows large spot motions. By replicating the array of spots slightly beyond the edge of the aperture, we reduce the chance for boundary phase dislocations in the reconstruction of the wave front. The method is especially suited to very large arrays. © 2004 Optical Society of America

OCIS codes: 010.1080, 010.7350, 100.2650, 100.5070, 150.0150, 220.4840.

1. INTRODUCTION

Wave-front sensing is an essential part of adaptive optics, vision optics, optical and silicon manufacturing, and more. Sensing is either of the wave front itself (point-diffraction interferometry), its gradient (shearing interferometry, Hartmann–Shack), or its Laplacian (curvature sensing). There are many variants of each of these sensors and others, and they are well described in many reviews and books.^{1–3} We examine here the problem of deciphering the data of a popular device, the Hartmann–Shack wave-front sensor (from here on, Hartmann sensor). In this device, the wave front is sampled by a regular grid of holes (Hartmann) or, more efficiently, lenslets (Hartmann–Shack). Local slopes cause the beams or the foci to move off their nominal positions by an amount proportional to the distance between the holes or the lenslets and the detector. An image of the whole distorted pattern of spots is then taken and analyzed to find the locations of each and every spot. Calculation is performed on a subregion around the expected location of each spot to find its centroid. To compare the wave front to some reference wave front, the locations of the reference spots are subtracted. These differences are broken into the x and y components of the gradient of the wave front to be further analyzed: Many times the wave front itself is reconstructed^{4–10}; in adaptive optics, the gradients serve as the input to the control loop of the wave-front corrector.

In some cases it is easier to calculate the gradients of the wave front using Fourier transforms instead of the centroid method just described. This applies especially to large arrays of spots and to large detectors, where fast Fourier transforms (FFT) have an advantage. In this algorithm,¹¹ the two sidelobes in the Fourier domain located at the frequency of the spots in both directions are each shifted to the origin, at the same time filtering out the rest of the transform. Two inverse Fourier trans-

forms follow whose phases are the two gradient components of the wave front. For adaptive optics, these sidelobes, even without inverse transform, are sufficient as an input to the control loop.¹¹ Unfortunately, this method suffers from edge effects: Data near the periphery of the aperture are corrupted if the boundary conditions cannot be specified in full. On the other hand, the filtering process in the Fourier domain (isolation of the sidelobes) smoothes the data and actually provides the wave front over the whole image domain, not just near the Hartmann spots. The Fourier scheme is also suited to cases where the aberrations are large and the Hartmann spots stray far from their original position. In these cases they are hard to find and identify.

2. DEMODULATION

We show that this modulation scheme can be performed without reverting to Fourier transforms, which simplifies the calculations significantly. For very large arrays, the saving translates to calculations that are more than an order of magnitude faster and save on cache requirements. Furthermore, all edge effects are now limited only to the last row or column of spots.

We would like to center and isolate the sidelobe of the Fourier space. Multiplying our image of the Hartmann pattern $I(\mathbf{r})$ by $\exp(-i\mathbf{q}_0 \cdot \mathbf{r})$ shifts (in Fourier space) the side lobe at \mathbf{q}_0 to the origin. Smoothing the complex result $I_0(\mathbf{r}) = I(\mathbf{r})\exp(-i\mathbf{q}_0 \cdot \mathbf{r})$ removes high Fourier components that are far from the origin. There are many possible ways of smoothing. The simplest one is the sliding average: The smoothed value is an average over a rectangular region $\mathbf{P} = (P_x, P_y)$:

$$I_0^s(x, y) = \frac{1}{P_x P_y} \sum_{j_x=-P_x/2}^{P_x/2} \sum_{j_y=-P_y/2}^{P_y/2} I_0(x + j_x, y + j_y). \quad (1)$$

The phase of $I_0^s(\mathbf{r})$ is the sought gradient; according to the direction of \mathbf{q}_0 it is the x or y component of the centroid motion. The Fourier transform of $I_0^s(\mathbf{r})$ has zeros on the grid $2\pi(m_x/P_x, m_y/P_y)$. The dimension of the smoothing rectangle \mathbf{P} can be chosen at will, so setting it to be the grid of the Hartmann spots ensures that all the other Fourier lobes except the desired one will be removed. Executing the smoothing twice would further reduce any leftovers.

What happens at the edges? We have Hartmann data only within some finite region \mathfrak{R} . The sliding average can be performed only inside it; near the edges, some of the smoothing rectangle \mathbf{P} may lie outside \mathfrak{R} , and other means will have to be adopted.

3. HARTMANN ANALYSIS

We expand significantly on the description of the Hartmann analysis⁹; a formal description is given in Appendix A. Here we skip the mathematical details and concentrate on the physics. We start by writing the expression for the irradiance function of the Hartmann pattern. Let $S(\mathbf{r}) = S(x, y)$ be an image of a single unperturbed Hartmann spot. [Although using \mathbf{r} alone is more succinct, we shall interchange it with (x, y) throughout the paper to clarify some points.] The unperturbed image is a rectangular grid of such spots

$$\begin{aligned} I_r(\mathbf{r}) &= \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} S(\mathbf{r} - \mathbf{r}_{n_x n_y}) \\ &\equiv \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} S(x - n_x P_x, y - n_y P_y), \end{aligned} \quad (2)$$

where the array is assumed infinite. When disturbed each spot $S(\mathbf{r})$ is displaced to a new position $\mathbf{r} + F\nabla\phi(\mathbf{r})$ where the shift is the product of the focal distance F and the local gradient of the wave front $\nabla\phi(\mathbf{r}) = \phi_x(\mathbf{r}) + \phi_y(\mathbf{r})$. The image is then

$$\begin{aligned} I(\mathbf{r}) &= \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} S[\mathbf{r} - \mathbf{r}_{n_x n_y} - F\nabla\phi(\mathbf{r}_{n_x n_y})] \\ &= \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} S\{[x - n_x P_x - F\phi_x(n_x P_x + n_y P_y)], \\ &\quad [y - n_y P_y - F\phi_y(n_x P_x + n_y P_y)]\}. \end{aligned} \quad (3)$$

We wish to find the gradient, which we assume to be varying slowly, and from it the wave front itself. In Fourier space, $s(\mathbf{q})$ is the Fourier transform of $S(\mathbf{r})$. For the unperturbed grid, the transform of the regularly spaced grid is a set of Dirac δ functions. For the perturbed grid, each δ function is shifted and spread a little.

$$\begin{aligned} I(\mathbf{r}) &= \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} S[\mathbf{r} - \mathbf{r}_{n_x n_y} - F\nabla\phi(\mathbf{r}_{n_x n_y})] \\ &\equiv \sum_{l_x=-\infty}^{\infty} \sum_{l_y=-\infty}^{\infty} s(\mathbf{q}_{l_x l_y}) \exp[-i\mathbf{q}_{l_x l_y} \cdot F\nabla\phi(\mathbf{r})] \exp(i\mathbf{q}_{l_x l_y} \cdot \mathbf{r}), \end{aligned} \quad (4)$$

where

$$\mathbf{q}_{l_x l_y} = (l_x q_x, l_y q_y) = (2\pi l_x / P_x, 2\pi l_y / P_y).$$

Of the many Fourier components present in approximations (4), the most prominent are the lowest ones. The noise level is rather constant, so that low- \mathbf{q} lobes have a better signal-to-noise ratio.¹² The $l_x = l_y = 0$ term does not depend on $\nabla\phi(\mathbf{r})$. The nearest two lobes are $\mathbf{q}_{10} = q_x = 2\pi/P_x$ and $\mathbf{q}_{01} = q_y = 2\pi/P_y$, which we term in general \mathbf{q}_0 .

Another important practical modification corresponds to variations of the amplitude of the spots across the Hartmann pattern, $A(\mathbf{r})$. These are caused by local scintillations—intensity changes in the Hartmann spots—and by the shape of the optical aperture—zero outside it, the average intensity inside. That is, the reference grid is not infinite any more, $I_r(\mathbf{r}) = \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} A(\mathbf{r}) S(\mathbf{r} - \mathbf{r}_{n_x n_y})$, and the distorted image is

$$I(\mathbf{r}) \equiv \sum_{l_x=-\infty}^{\infty} \sum_{l_y=-\infty}^{\infty} s(\mathbf{q}_{l_x l_y}) A(\mathbf{r}) \exp[-i\mathbf{q}_{l_x l_y} \cdot F\nabla\phi(\mathbf{r})].$$

As shown in Appendix A, two conditions apply: First, we are limited to the inside of the region \mathfrak{R} where $A(\mathbf{r}) > 0$. Second, both $F\nabla\phi(\mathbf{r})$ and $A(\mathbf{r})$ should vary slowly over the region \mathbf{P} near the spots. The aperture $A(\mathbf{r})$ should be much wider than the interspot distance. In other words, the Fourier transform of $A(\mathbf{r})$ is narrow, not wide enough to mix nearby spikes in the Fourier domain:

$$\nabla[\mathbf{q}_0 \cdot F\nabla\phi(\mathbf{r})] \ll 1, \quad (5a)$$

$$|\nabla A(\mathbf{r})|^2 / A^2(\mathbf{r}) \ll q_0^2. \quad (5b)$$

The second condition arises because we have to assume that the aperture transform is not wide enough to mix nearby spikes in the Fourier domain. In general, if $A(\mathbf{r})$ changes appreciably only over distances larger than some R , then its highest Fourier components are $Q \approx 2\pi/R$. The dominant frequency of the function $A(\mathbf{r})$ is $\langle q^2 \rangle = \langle q^2 A(q)^2 \rangle / \langle A(q)^2 \rangle = |\nabla A(\mathbf{r})|^2 / A^2(\mathbf{r})$. This is equivalent to requiring that $R \gg P$, the distance between spots, or that there are many spots in the image, $q_0 \gg Q$.

Phase discontinuities can occur if the phase fluctuations do not obey relations (5). This is true at the periphery of the aperture, and in the Fourier method¹⁰ such errors may propagate deep into the aperture. Here the propagation depth is limited to the size of the smoothing kernel, which is the pitch of the lenslets. A way to reduce this edge effect even further is to extend the results beyond the boundary. We know that the integral of the phase ϕ should be zero around every closed loop, including those loops straddling the edge. This can be achieved by requiring that the ϕ_x will be constant beyond the left and right edges of the aperture and ϕ_y , beyond the top and bottom edges. A number of ways were devised to extend the gradients beyond the edge,¹⁰ and here we preferred to operate on the raw Hartmann pattern itself. We copied the row or column of spots just inside the boundary beyond the edge. A swath of width P was copied to all four sides of the round aperture. Since this makes two copies at some pixels from the horizontal and

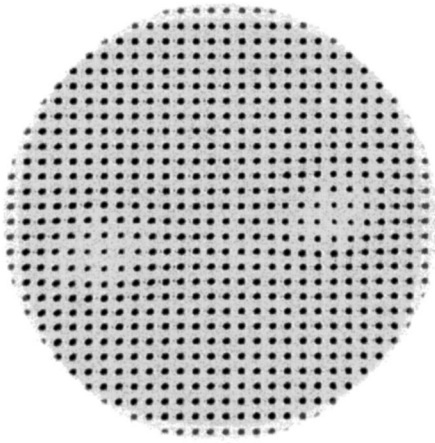


Fig. 1. Extending the Hartmann pattern by one row of spots on each side reduces phase dislocations. The data are shifted by one period to each of the four sides of the original and these four extensions are averaged, then the original is plugged back in. Note also the reduced amplitude near the two highly aberrated regions caused by scintillation.

vertical duplicates, the final values were averaged between them (Fig. 1). This essentially solved the phase-dislocation problem. However, this extension method is limited to duplicates at integer numbers of pixels, whereas the pitch is not necessarily so, leading to small errors at the edge proper. For undersampled arrays or noninteger pitch, an interpolation might be required.

4. PROCEDURE

We list the algorithmic description of the processing stages, starting with extension of the images by one layer of spots to each side of \mathfrak{R} , the region where data exist (step 6a), and plugging the image back into \mathfrak{R} (step 6b). Then the x demodulation is performed. First, we shift the phase (step 6c), then apply a sliding average (smoothing filter), first in x (step 6d) and then in y (step 6e). Further smoothing is achieved in a second pass, first in x (step 6f) and then in y (step 6g). Finally, the phase gradient is calculated (step 6h). The demodulation steps for the y direction are similar (but not listed). Analysis of these stages is given in Appendix B.

$$I^e(x, y) = \frac{1}{2}[I(x - P_x/2, y) + I(x + P_x/2, y) + I(x, y - P_y/2) + I(x, y + P_y/2)], \quad (6a)$$

$$I^e(\mathfrak{R}) = I^{0e}(\mathfrak{R}), \quad (6b)$$

$$I^c(x, y) = I^e(x, y)\exp(-2\pi ix/P_x), \quad (6c)$$

$$I^{1x}(x, y) = \sum_{j=0}^{P_x-1} I^c(x + j - P_x/2, y)/P_x, \quad (6d)$$

$$I^{1y}(x, y) = \sum_{j=0}^{P_y-1} I^{1x}(x, y + j - P_y/2)/P_y, \quad (6e)$$

$$I^{2x}(x, y) = \sum_{j=0}^{P_x-1} I^{1y}(x + j - P_x/2, y)/P_x, \quad (6f)$$

$$I^{2y}(x, y) = \sum_{j=0}^{P_y-1} I^{2x}(x, y + j - P_y/2)/P_y, \quad (6g)$$

$$\phi_x(x, y) = \arg[I^{2x}(x, y)]P_x/2\pi F. \quad (6h)$$

One could do with a single pass of sliding average, i.e., without stages (6f) and (6g), but the noise characteristics of the double-pass smoothing are superior. The double-pass sliding average can be considered a triangular filter because of the shape of its kernel. In Appendix B we consider also another smoothing scheme, which is noisier but (sometimes) quicker. A practical question may arise, for noninteger pitch, how to average over, say, 13.7 pixels. In this case one should sum 13 pixels and add 0.7 of the 14th one, and divide by 13.7. The results in the appendixes are valid for any (real) pitch.

To test these demodulation methods, we used the same set of Hartmann data taken from a known sample. We processed it by three methods: (a) convolution with the

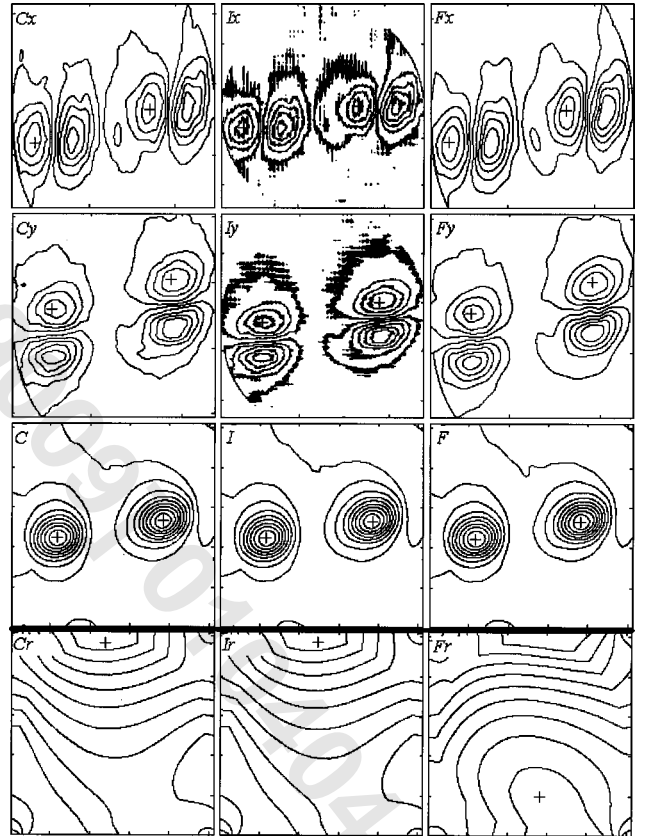


Fig. 2. Comparison of results from three Hartmann demodulation techniques using the image from Fig. 1. Top row, x gradients; second row, y gradients; third row, integrated phases; bottom row, reference phase. Left column, convolution with a kernel of size \mathbf{P} ; center column, smoothing (single pass) as in Eqs. (6) or (B7); right column, Fourier analysis. Positive contours are marked by a central $+$. The elevation of the two features is $1.2 \mu\text{m}$ (third row, contour spacing $0.15 \mu\text{m}$), to be compared with the much larger $6 \mu\text{m}$ for the references (bottom row, contour spacing $0.77 \mu\text{m}$). The diameter of the round aperture is visible in some images. Note how different is the Fourier reference (bottom right), which results from a shift of the lobe to the center by integer frequencies. As a result, a tilt is added to the Fourier reference that also appears in the results and hence subtracts perfectly.

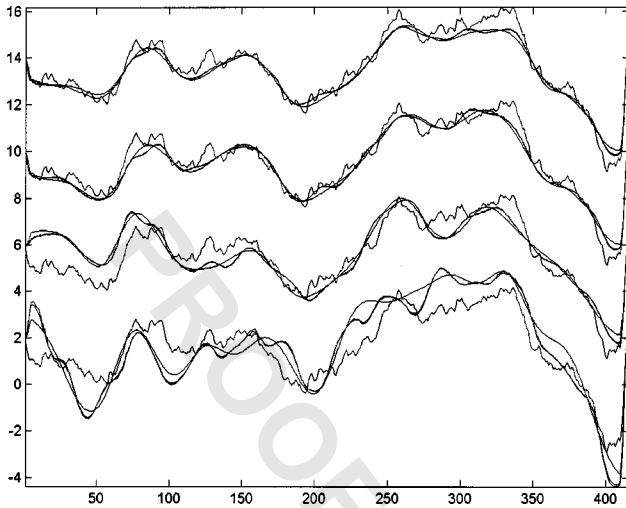


Fig. 3. Simulation of cuts across the round aperture: Given a wave front (jagged curve), it is converted into a Hartmann pattern with lenslet pitch of 11 pixels. Then Poisson noise at the average levels of 16, 64, 256, and 1024 photons per lenslet was added (four groups starting from the bottom, respectively). The results are always smoother because of the finite size of the lenslets. In addition, the smoothest result is the Fourier method, then the convolution, and, very close to it, the single-pass smoothing.

triangular filter, (b) the smoothing filter listed in Eqs. (6), and (c) a direct Fourier deconvolution.¹¹ The results were readily phase-unwrapped from the center of the aperture outward. As a last stage of the processing, we reconstructed the phase from its gradients by using the least-squares Fourier fit^{9,10} (Fig. 2). Since our data were immune from phase dislocations as a result of the duplication of the boundary, we did not encounter any phase wrapping problems. All results are very close to each other and to the measured phase as obtained by different means, such as interferometry, all within a few percent. Results produced with different Hartmann arrays at different pitches \mathbf{P} and different focal lengths F are also similar to within a few percent.

All measurements included a reference (or calibration) wave front whose phase was subtracted from the final result, where the subtraction was effected as a phasor. Toward this end we calculated the wave fronts of the object and of the reference from their corresponding images by using the method described above. However, we did not peel out the phases from the results, as in stage (6h), to subtract them later. Instead, we multiplied the two complex quantities $I_{\text{obj}}^x(\mathbf{r}) = A(\mathbf{r})\exp[-iF\phi_{\text{obj}}^x(\mathbf{r})]$ and $I_{\text{ref}}^x(\mathbf{r}) = A(\mathbf{r})\exp[-iF\phi_{\text{ref}}^x(\mathbf{r})]$ from the previous stage (6g) as follows:

$$I_{\text{obj}}^x(\mathbf{r})[I_{\text{ref}}^x(\mathbf{r})]^* = [A(\mathbf{r})]^2 \exp\{-iF[\phi_{\text{obj}}^x(\mathbf{r}) - \phi_{\text{ref}}^x(\mathbf{r})]\}, \quad (7)$$

and similarly for the y derivative. The phase of the result is the calibrated gradient. The principal advantage of using this variation is the relaxation of the phase wrapping problem. To get the complex phases of both $I_{\text{obj}}^x(\mathbf{r})$ and $I_{\text{ref}}^x(\mathbf{r})$ without discontinuities, the changes in both phase derivatives must be small compared to 2π [relation (5a)]. Using the normal method, this condition applies separately to both the reference and the object. In prac-

tice, it limits the amplitude of phases of the wave front that can be measured. With the method outlined in Eq. (7), this condition applies only to the *difference* in the wave fronts, and the range of measurable wave fronts may be extended arbitrarily by use of suitable references.

Indeed, these reference images were very different among the various demodulation schemes (Fig. 2, bottom row). This is because it is very difficult to shift the Fourier sidelobes to the center with subpixel accuracy, these being highly variable, complex values. So shifting by integer pixels results in residual (subpixel shift) errors which lead to some tip and tilt in the gradients. These gradients integrate to a curvature in the wave front whose values are known and can be removed (Fig. 2). However, shifting both the reference and the measured phase gradients by the same frequency in the Fourier domain [Eq. (7)] totally removed this curvature effect in the final result. The process was repeated for simulated input, and the results are shown in Fig. 3 for different noise levels.

5. NET EXAMPLE

A mosquito net is a regular grid of threads with holes in between. Each cell is composed of the bounding threads and the hole. Defects in the net change the regular grid. A map of the displacement of each cell (relative to the unperturbed grid) may be used to locate the defects during the manufacturing process. One could consider each cell to be a Hartmann point and apply the above procedure. The displacement is exactly the phase, up to a normalization constant. The defects are most prominent on maps of local changes in the displacement. For on-line quality control, processing time is of paramount importance: Processing should be quicker than the production itself.

We tested our algorithm also on such a net. The original image was 2048×2048 pixels with cell size of 5–6 pixels, or approximately 130,000 cells per image. The threads appeared as thin, single-pixel-width lines, and there were prominent moiré and aliasing effects in the image. Out of the net, a 512×512 map (or matrix) of the “displacement derivative” was produced by using the above method. There was no need for better resolution of the output. This displacement map was further processed to locate defects, if any.

We used the mosquito net as an example of a large-format problem we set out to solve. We discussed a single-smoothing (block sum) or a double-smoothing process (triangular sums). In practice, it was found that triple smoothing yielded an even lower noise level. The times we quote below refer to smoothing three times in the x direction followed by smoothing three times in the y direction. Both directions were always required, because in practice \mathbf{q}_0 was neither purely horizontal nor purely vertical, but always a combination of both.

The whole process—from input image to derivative map—took 0.053 s on a 2.2-GHz Pentium 4 computer. Producing a displacement map instead of derivatives took about the same time. Processing time was linear with

the image dimensions and independent of the cell dimensions. That is why so many cells were squeezed into a single image. Aliasing did not affect the results as long as the point-spread function of the optical system was larger than the physical pixel's dimensions.

We wish to compare this time with the FFT processing time. In that approach the following steps are taken¹¹: (a) Zero-pad the array to double its linear dimensions to avoid aliasing, (b) Fourier transform the two-dimensional image, (c) shift the Fourier x sidelobe to the center, (d) remove all the other lobes by multiplying the transform with an apodizing function, (e) inverse Fourier transform, and (f)–(h) repeat steps (c)–(e) for the y sidelobe.

A two-dimensional FFT on an image of size $M \times N$ requires $5(M \times N)\log_2(M \times N)$ operations. Because of stage (a) the array size is $4 \times M \times N$ and we get $20(M \times N)[2 + \log_2(N \times M)]$ for each of steps (a), (d), and (h). Steps (c) and (f) are one operation each and (d) and (g) are $4(M \times N)$ operations each. Thus we get $\Xi \approx M \times N[8 + 120 + 60\log_2(N \times M)]$. With $N = M = 2048$, $\Xi \approx 2 \times 10^9$. The best available two-dimensional FFT package reaches 1.5×10^9 operations/s on the same computer in a program written in C and utilizing MMX operations, or 1.3 s. The whole process will last ≈ 4 s, which is 76 times slower than the smoothing approach.

The huge disparity in speed results from the fact that the simple shift operations required for the sliding average in the smoothing operation [Eqs. (B2)] in both x and y directions result in only $10(M \times N)$ operations (zero padding is not necessary). Performing this filter l times requires $10l(M \times N)$ operations. Each direct triangular convolution of size P in both directions results in a heavier $\approx PM \times PN$ operations.

Finally, we compared the processing times on the same Hartmann data, but at different array sizes, by using MATLAB, which is optimized for matrix and FFT operations. We smoothed the original 512^2 array down to 256^2 and 128^2 , and measured the demodulation time alone for these three sizes. We found that the Fourier method was the fastest for the smallest array by a factor of four when compared with the other methods but was comparable with them for the medium array and slower for the largest one. This is explained by the limited memory size of the computer, which requires caching of large arrays when performing the three Fourier transforms. The smoothing method was the fastest for the largest array by a factor of two when compared with the other methods. The balance might shift with different computers and with optimized code other than the commercial one we used.

The traditional centroiding procedure can also be viewed as a convolution process. This is especially clear for the case of exactly two pixels allocated for each spot, where one adds the first and subtracts the second pixel to find the centroid along each dimension. If this is done in parallel for the whole array, the similarity to our algorithm emerges immediately. Regarding noise propagation in this algorithm, we suspect that it would be similar to the centroid rather than the FFT algorithm¹¹ in that the noise spreads more in the vicinity of its origin rather than over the whole Fourier domain. Further studies need to be undertaken to verify this point.

APPENDIX A

The purpose of this Appendix is to provide a more formal and detailed description of the demodulation process. We consider a rectangular region of space (an image) $L_x \times L_y$. Inside this area is a rectangular grid of Hartmann spots. The centers of the unperturbed spots are located at positions $\mathbf{r}_{n_x n_y} = (n_x P_x, n_y P_y)$. L_x and L_y are chosen so as to contain exactly M_x columns and M_y rows so that $L_x = M_x P_x$, $L_y = M_y P_y$.

Let $S(\mathbf{r})$ be the image of a single unperturbed Hartmann spot centered on $(0, 0)$. The image of a displaced spot centered at $\mathbf{r}_{n_x n_y}$ is $S(\mathbf{r} - \mathbf{r}_{n_x n_y})$; the Fourier transform of $S(\mathbf{r})$ is $s(\mathbf{q})$. Stepping into our limited world $0 \leq x < L_x$; $0 \leq y < L_y$, we define a cyclic $CS(\mathbf{r})$ beyond it:

$$CS(x, y) = \sum_{i_x=-\infty}^{\infty} \sum_{i_y=-\infty}^{\infty} S(x + i_x L_x, y + i_y L_y). \quad (\text{A1})$$

$S(\mathbf{r})$ is localized, dropping to zero at distances much shorter than L_x or L_y . Actually, they would usually be shorter than P_x or P_y for distinct Hartmann spots to be visible. Thus, this extension influences only spots very close to the edges of the image. Such spots are negligible, as shown later.

$CS(\mathbf{r})$ has a translational symmetry and can be represented as a discrete Fourier sum (the Fourier transform of a set of Dirac δ functions),

$$CS(\mathbf{r}) = \frac{4\pi^2}{L_x L_y} \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(q_x, q_y) \exp(i\mathbf{q} \cdot \mathbf{r}), \quad (\text{A2})$$

where $(q_x, q_y) = 2\pi(m_x/L_x, m_y/L_y)$. Note that $CS(\mathbf{r})$ is a continuous function and the Fourier transform is discrete. Consider now the combined image of the grid of all spots within our $L_x \times L_y$ region. With the exception of spots near the edges, where there is some error, we can write the image as

$$I_r(\mathbf{r}) = \sum_{n_x, n_y} S(\mathbf{r} - \mathbf{r}_{n_x n_y}) \Rightarrow \sum_{n_x, n_y} CS(\mathbf{r} - \mathbf{r}_{n_x n_y}), \quad (\text{A3})$$

$$I_r(x, y) = \sum_{n_x=1}^{M_x} \sum_{n_y=1}^{M_y} CS(x - n_x P_x, y - n_y P_y). \quad (\text{A4})$$

For a perfect grid of spots we have for every x and y a stronger translational symmetry than for CS alone: $I_r(x + P_x, y) = I_r(x, y + P_y) = I_r(x + P_x, y)$. The discrete Fourier transform has fewer terms:

$$I_r(\mathbf{r}) = \frac{4\pi^2}{P_x P_y} \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(q_x, q_y) \exp(i\mathbf{q} \cdot \mathbf{r}), \quad (\text{A5})$$

where now $(q_x, q_y) = 2\pi(m_x/P_x, m_y/P_y)$. Equation (A5) is independent of the dimensions of the region L_x and L_y . This region was introduced to simplify the tran-

sition from Fourier integrals to Fourier sums. It is a finite region of very large dimensions so that the effect of the difference between $S(\mathbf{r})$ and $CS(\mathbf{r})$ is negligible.

When the wave front is disturbed, each spot is displaced by $F\nabla\phi(\mathbf{r})$. The modified image is

$$\begin{aligned} I(\mathbf{r}) &= \sum_{n_x, n_y} S[\mathbf{r} - \mathbf{r}_{n_x n_y} - F\nabla\phi(\mathbf{r}_{n_x n_y})] \\ &\Rightarrow \sum_{n_x, n_y} CS[\mathbf{r} - \mathbf{r}_{n_x n_y} - F\nabla\phi(\mathbf{r}_{n_x n_y})] \\ &\cong \sum_{n_x, n_y} CS[\mathbf{r} - \mathbf{r}_{n_x n_y} - F\nabla\phi(\mathbf{r})]. \end{aligned} \quad (\text{A6})$$

The approximation $\nabla\phi(\mathbf{r}) = \nabla\phi(\mathbf{r}_{n_x n_y})$ is valid if $\nabla\phi(\mathbf{r})$ changes slowly over the region of $S(\mathbf{r})$. Since $S(\mathbf{r})$ is very localized, the approximation holds. Later, we relax the condition by using a second iteration stage. $CS(\mathbf{r} - \mathbf{r}_{n_x n_y} - \mathbf{M})$ differs from zero only at (or near) $\mathbf{r} = \mathbf{r}_{n_x n_y} - \mathbf{M}$, with $\mathbf{M} = \nabla\phi(\mathbf{r})$ evaluated at $\mathbf{r} = \mathbf{r}_{n_x n_y}$. The approximation replaces this discrete set of \mathbf{M} values with the continuous function $\nabla\phi(\mathbf{r})$ evaluated at \mathbf{r} . Thus, $\mathbf{M}(\mathbf{r}_{n_x n_y})$ is replaced with $\mathbf{M}(\mathbf{r}_{n_x n_y} + \mathbf{M})$. The conditions of relations (5) requiring phase smoothness must be met for these approximations to hold. We also include the amplitude of the Hartmann pattern $A(\mathbf{r})$. Substituting into Eq. (A5) we get

$$\begin{aligned} I_r(\mathbf{r}) &= A(\mathbf{r}) \frac{4\pi^2}{P_x P_y} \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(q_x, q_y) \\ &\quad \times \exp(i\mathbf{q} \cdot \mathbf{r}) \exp[-i\mathbf{q} \cdot \nabla\phi(\mathbf{r})]. \end{aligned} \quad (\text{A7})$$

The most prominent components are $\mathbf{q}_{10} = (2\pi/P_x, 0)$ and $\mathbf{q}_{01} = (0, 2\pi/P_y)$. To extract the gradient $\phi_x(\mathbf{r})$ we multiply $I(\mathbf{r})$ by $\exp(-i2\pi x/P_x)$ or $\exp(-i\mathbf{q}_{10} \cdot \mathbf{r})$. This shifts the \mathbf{q}_{10} component into $\mathbf{q} = 0$. Next, we smooth the function with a weight function \mathbf{W} to remove all other Fourier components. A simple weight function is the sliding average $W(x, y) = 1/(P_x P_y)$ in the region $-P_x/2 < x < P_x/2$, $-P_y/2 < y < P_y/2$ and zero elsewhere. Other functions may be used. If the variations of the intensity $A(\mathbf{r})$ and phase gradient $\nabla\phi(\mathbf{r})$ over the small integration region are neglected, all frequency terms vanish except at \mathbf{q}_{10} , and we get $I_{10}(\mathbf{r}) = I(\mathbf{r}) \exp(-i\mathbf{q}_{10} \cdot \mathbf{r})$. To show this more precisely, we see that the smoothed version is Eq. (1):

$$\begin{aligned} I_{10}^S(x, y) &= (P_x P_y)^{-1} \int_{-P_x/2}^{P_x/2} dx' \int_{-P_y/2}^{P_y/2} dy' \\ &\quad \times I_{10}(x + x', y + y') \\ &= P^{-2} \int \int_P d^2\mathbf{r}' I_{10}(\mathbf{r} + \mathbf{r}'), \end{aligned} \quad (\text{A8})$$

which we write as

$$I_{10}^S(\mathbf{r}) = \frac{4\pi^2}{P_x P_y} \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(\mathbf{q}) V_{10}(\mathbf{q}, \mathbf{r}), \quad (\text{A9})$$

where

$$\mathbf{q} = (q_x, q_y) = 2\pi(m_x/P_x, m_y/P_y),$$

$$\begin{aligned} V_{10}(\mathbf{q}, \mathbf{r}) &= P^{-2} \int \int_P d^2\mathbf{r}' A(\mathbf{r} + \mathbf{r}') \\ &\quad \times \exp[i(\mathbf{q} - \mathbf{q}_{10}) \cdot (\mathbf{r} + \mathbf{r}')] \\ &\quad - i\mathbf{q} \cdot F\nabla\phi(\mathbf{r} + \mathbf{r}'). \end{aligned} \quad (\text{A10})$$

If $A(\mathbf{r} + \mathbf{r}')$ and $\phi(\mathbf{r} + \mathbf{r}')$ change little around \mathbf{r} , we may replace them by their averages around the r and take them out of the integral. The integration then yields delta functions:

$$\begin{aligned} V_{10}(\mathbf{q}, \mathbf{r}) &\cong \bar{A}(\mathbf{r}) B(\mathbf{q} - \mathbf{q}_{10}) \\ &\quad \times \exp[i(\mathbf{q} - \mathbf{q}_{10}) \cdot \mathbf{r} - i\mathbf{q} \cdot \overline{F\nabla\phi(\mathbf{r})}], \end{aligned} \quad (\text{A11})$$

where

$$\begin{aligned} B(\mathbf{q} - \mathbf{q}_{10}) &= (P_x P_y)^{-1} \int_{-P_x/2}^{P_x/2} dx' \int_{-P_y/2}^{P_y/2} dy' \\ &\quad \times \exp\{2\pi i[x'(m_x - 1)/P_x + y'm_y/P_y]\} \\ &= \delta(m_x - 1)\delta(m_y), \end{aligned} \quad (\text{A12})$$

and we have

$$I_{10}^S(\mathbf{r}) = 4\pi^2 (P_x P_y)^{-1} \bar{A}(\mathbf{r}) \exp[-i\mathbf{q}_{10} \cdot \overline{F\nabla\phi(\mathbf{r})}]. \quad (\text{A13})$$

The errors in Eq. (A13) are caused by neglecting variation of $A(\mathbf{r})$ and $F\nabla\phi(\mathbf{r})$ over the small integration rectangle \mathbf{P} . The magnitude of $I^S(\mathbf{r})$ is proportional to the average amplitude, which we denote $\bar{A}(\mathbf{r})$, and its phase is the average gradient denoted by $\overline{F\nabla\phi(\mathbf{r})}$. Other errors could originate from the approximation in Eq. (A5). We may reduce the errors further by removing most of the variations in a second iteration. In the first iteration we calculate an approximate solution $\bar{A}(\mathbf{r})$ and $\overline{F\nabla\phi(\mathbf{r})}$. We then use those values to modify, or morph, the intensity distribution $I(\mathbf{r})$ prior to the smoothing into its "original unperturbed form." Thus, we are left with a very small perturbation to a perfect Hartmann grid. Then we calculate, by using the morphed $I(\mathbf{r})$ (which is almost a perfect grid) and the method above, the small residual correction $dA(\mathbf{r})$ and $dF\nabla\phi(\mathbf{r})$:

$$I^m(\mathbf{r}) = I[\mathbf{r} + \overline{F\nabla\phi(\mathbf{r})}] / \bar{A}(\mathbf{r}). \quad (\text{A14})$$

This correction is valid only where $\bar{A}(\mathbf{r}) \neq 0$, namely inside the aperture. Where $\bar{A}(\mathbf{r}) \approx 0$, we substitute values from the vicinity, as in Eqs. (6a) and (6b).

The modified, or morphed, integral term $B(\mathbf{q})$ in Eq. (A12) is, after removal of the zeroth-order amplitude and phase,

$$\begin{aligned} B^m(\mathbf{q} - \mathbf{q}_{10}) &= P^{-2} \int \int_P d^2\mathbf{r}' \exp[i(\mathbf{q} - \mathbf{q}_{10}) \cdot (\mathbf{r} + \mathbf{r}')] \\ &\quad \times [\exp(-i\mathbf{q} \cdot \delta\phi + \delta A)], \end{aligned} \quad (\text{A15})$$

where $\delta\vartheta = \overline{F\nabla\phi(\mathbf{r} + \mathbf{r}')} - \overline{F\nabla\phi(\mathbf{r})}$ and $\delta A = \log A(\mathbf{r} + \mathbf{r}') - \log A(\mathbf{r})$.

With the introduction of morphing,

$$I_{10}^m(\mathbf{r}) = 4\pi^2 P^{-2} dA(\mathbf{r}) \exp[-i\mathbf{q}_{10} \cdot \nabla F d\phi(\mathbf{r})], \quad (\text{A16})$$

where

$$d\phi(\mathbf{r}) = \phi(\mathbf{r}) - \bar{\phi}(\mathbf{r}) \approx 0, \quad dA(\mathbf{r}) = A(\mathbf{r})/\bar{A}(\mathbf{r}) \approx 1$$

inside the aperture, and hence $d \log[A(\mathbf{r})] \approx 0$. Since the terms themselves are small, the errors caused by their variations are much smaller.

Along the boundaries of the aperture there is a strip of width $P/2$ where we do not have information: The amplitude A drops to zero here. The simplest approximation is to assume that the gradient is constant and use the closest known value [Eqs. (6a) and (6b) and Fig. 1]. For a round aperture of radius R we take $\nabla\phi(r) = \nabla\phi(r - P)$; $R < r < R + P$.

APPENDIX B

We show that the smoothing filter we have applied to the data is equivalent to removal of higher-order lobes, and suggest an alternative filter. We first move the sidelobe to the origin in the Fourier domain, which is equivalent to a convolution with $\delta(\mathbf{q} - \mathbf{q}_0)$. This simply amounts to obtaining the product $I_0(\mathbf{r}) = I(\mathbf{r}) \exp(-i\mathbf{q}_0 \cdot \mathbf{r})$. Next, we wish to remove the rest of the grid and start with the columns. Removing the grid elements amounts to multiplication in the frequency domain of each element by $\cos^2(qP_x/4) = [1 + \cos(qP_x/2)]/2$, thus zeroing lobes at $qP_x/4 = (2n + 1)\pi/2$, or $q = \pm q_x, \pm 3q_x, \pm 5q_x, \dots$. But multiplying with a squared cosine function in the frequency domain is equivalent to averaging the original function at both sides of the original $\frac{1}{2}I_0(\mathbf{r} - P_x/4) + \frac{1}{2}I_0(\mathbf{r} + P_x/4)$ and then repeating it, which in turn is equivalent to applying the filter $\frac{1}{2}I_0(\mathbf{r}) + \frac{1}{4}I_0(\mathbf{r} - P_x/2) + \frac{1}{4}I_0(\mathbf{r} + P_x/2)$. Similarly, removing the more distant lobes at $q = \pm 2q_x, \pm 6q_x, \pm 10q_x, \dots$ is achieved by averaging again $\frac{1}{2}I_0(\mathbf{r}) + \frac{1}{4}I_0(\mathbf{r} - P_x/4) + \frac{1}{4}I_0(\mathbf{r} + P_x/4)$, and so on. In general, averaging $\frac{1}{2}I_0(\mathbf{r}) + \frac{1}{4}I_0(\mathbf{r} - P_x/2j) + \frac{1}{4}I_0(\mathbf{r} + P_x/2j)$ gets rid of lobes at $q = \pm jq_x, \pm 3jq_x, \pm 5jq_x, \dots$.

This smoothing (averaging of a point with values at its neighbors) also influences somewhat the Fourier components very close to the origin: These are multiplied by $\cos^2(qP_x/4)$ (for the first filter). If this small smoothing matters, we can correct it by using a more sophisticated filter such as $\cos^2(qP_x/4)[2 - \cos^2(qP_x/4)]$. Its application in real space is rather simple and requires applying each filter twice. The filtered result is:

$$\begin{aligned} F(\mathbf{r}) &= \frac{1}{2}I_0(\mathbf{r}) + \frac{1}{4}I_0(\mathbf{r} - P_x/4) + \frac{1}{4}I_0(\mathbf{r} + P_x/4), \\ G(\mathbf{r}) &= \frac{1}{2}F(\mathbf{r}) + \frac{1}{4}F(\mathbf{r} - P_x/4) + \frac{1}{4}F(\mathbf{r} + P_x/4), \\ T_f(\mathbf{r}) &= 2F(\mathbf{r}) - G(\mathbf{r}). \end{aligned} \quad (\text{B1})$$

If we define the operation $\Theta_j C(\mathbf{r}) \equiv \frac{1}{2}C(\mathbf{r} - P/4j) + \frac{1}{2}C(\mathbf{r} + P/4j)$, then Eq. (B1) can be written as $T_f = 2\Theta_1 T - \Theta_1 \Theta_1 T$. If filters are applied in a sequence

at different pitches, such as $P_x/4, P_x/8$, then this repeated filtering will be $\Theta_2 \Theta_1 T$ and corrected filtering [generalized Eq. (B1)] will be $2\Theta_2 \Theta_1 T - \Theta_2 \Theta_1 \Theta_2 \Theta_1 T$. The number of repetitions and corrections can be decided according to the desired accuracy and the time allotted to the calculation. We spell out the result which seems to satisfy both requirements:

$$\begin{aligned} I^e(x, y) &= I(x, y) \exp(-2\pi i x / P_x), \\ I^{1y}(x, y) &= \frac{1}{2}I^e(x, y) + \frac{1}{4}I^e(x, y - P_y/2) \\ &\quad + \frac{1}{4}I^e(x, y + P_y/2), \\ I^{2y}(x, y) &= \frac{1}{2}I^{1y}(x, y) + \frac{1}{4}I^{1y}(x, y - P_y/4) \\ &\quad + \frac{1}{4}I^{1y}(x, y + P_y/4), \\ I^{1x}(x, y) &= \frac{1}{2}I^{2y}(x, y) + \frac{1}{4}I^{2y}(x - P_x/2, y) \\ &\quad + \frac{1}{4}I^{2y}(x + P_x/2, y), \\ I^{2x}(x, y) &= \frac{1}{2}I^{1x}(x, y) + \frac{1}{4}I^{1x}(x - P_x/4, y) \\ &\quad + \frac{1}{4}I^{1x}(x + P_x/4, y), \\ \phi_x(x, y) &= \arg[I^{2x}(x, y)] P_x / 2\pi F, \end{aligned} \quad (\text{B2})$$

and similarly for the y direction. This is a repetition of Eqs. (6) without the preliminary boundary extension stage. Again, finer results can be obtained by repeating with steps of $1/8$ the pitch. Alternatively, the results can be smoothed, but the smoothing filter will now be much narrower than the filter above.

A careful inspection of this operation shows it is rather similar to a convolution with a triangular filter with half-width of the Hartmann pitch. Indeed, a triangular convolution is exactly the result of averaging with the full series of filters above—with $P_x/2, P_x/4, P_x/8, P_x/16, P_x/32$, and so forth up to infinity. To see this, examine the Fourier transform of the series of filters

$$\begin{aligned} \cos(q)\cos(q/2)\cos(q/4) &= [\cos(7q/8) + \cos(5q/8) \\ &\quad + \cos(3q/8) + \cos(q/8)]/4, \end{aligned}$$

or, more generally,

$$\begin{aligned} C_1(q) &= \prod_{k=0}^N \cos(2^k q / 2^N) \\ &= 2^{-N} \sum_{m=1}^{2^N} \cos[(2m - 1)q / 2^{N+1}], \end{aligned} \quad (\text{B3})$$

whose transform is a simple rectangular filter

$$c_1(x) = 2^{-N-1} \sum_{m=-2^{N+1}}^{2^N} \delta[x - (2m - 1)/2^{N+1}]. \quad (\text{B4})$$

Squaring $C_1(q)$ we get

$$\begin{aligned}
C_2(q) &= \left\{ 2^{-N} \sum_{m=1}^{2^N} \cos[(2m-1)q/2^{N+1}] \right\}^2 \\
&= 2^{-2N-2} \sum_{m,k=-2^{N+1}}^{2^N} \exp[i(2m+2k-2)q/2^{N+1}] \\
&= 2^{-2N-2} \sum_{p=-2^{N+1}}^{2^{N+1}} [2^{N+1} - |p|] \exp[ipq/2^N], \quad (\text{B5})
\end{aligned}$$

whose transform is now a triangular filter

$$c_2(x) = 2^{-2N-2} \sum_{p=-2^{N+1}}^{2^{N+1}} (2^{N+1} - |p|) \delta(x - p/2^N). \quad (\text{B6})$$

The convolution is performed first in the x direction, then in the y direction with this triangular filter (this is somewhat faster than a full, two-dimensional convolution). The disadvantage of this filter is that if the camera samples the Hartmann pattern at high resolution, the filter is wide and the convolution rather long.

ACKNOWLEDGMENTS

This work emerged from research in astronomical and ocular adaptive optics. It was supported in part by the Israeli Ministry of Science under the Adaptive Optics for the Eye program. Another part was supported by the European Community Research and Training Network

SHARP-EYE. Thanks are due to the reviewers, whose comments helped to improve this presentation.

REFERENCES

1. D. Malacara, ed., *Optical Shop Testing* (Wiley, New York, 1978).
2. R. K. Tyson, *Principles of Adaptive Optics*, 2nd ed. (Academic, New York, 1998).
3. R. K. Tyson, ed., *Adaptive Optics Engineering Handbook* (Marcel Dekker, New York, 2000).
4. D. L. Fried, "Least-squares fitting of a wave-front distortion estimate to an array of phase-difference measurements," *J. Opt. Soc. Am.* **67**, 370–375 (1977).
5. R. H. Hudgin, "Wave-front reconstruction for compensated imaging," *J. Opt. Soc. Am.* **67**, 375–378 (1977).
6. W. H. Southwell, "Wave-front estimation from wave-front slope measurement," *J. Opt. Soc. Am.* **70**, 998–1006 (1980).
7. F. Roddier and C. Roddier, "Wavefront reconstruction using iterative Fourier transforms," *Appl. Opt.* **30**, 1325–1327 (1991).
8. C. Roddier and F. Roddier, "Wave-front reconstruction from defocused images and the testing of ground-based optical telescopes," *J. Opt. Soc. Am. A* **10**, 2277–2287 (1993).
9. K. R. Freischlad and C. L. Koliopoulos, "Modal estimation of a wave front from difference measurements using the discrete Fourier transform," *J. Opt. Soc. Am. A* **3**, 1852–1861 (1986).
10. L. A. Poyneer, D. T. Gavel, and J. M. Brase, "Fast wave-front reconstruction in large adaptive optics systems with use of the Fourier transform," *J. Opt. Soc. Am. A* **19**, 2100–2111 (2002).
11. Y. Carmon and E. N. Ribak, "Phase retrieval by demodulation of a Hartmann–Shack sensor," *Opt. Commun.* **215**, 285–288 (2003).
12. E. N. Ribak, "Separating atmospheric layers in adaptive optics," *Opt. Lett.* **28**, 613–615 (2003).