Supplementary Material: Megapixel Adaptive Optics

CONGLI WANG, QIANG FU, XIONG DUN, and WOLFGANG HEIDRICH,
King Abdullah University of Science and Technology, Saudi Arabia

S.1 WAVEFRONT SOLVER DETAILS

Derivations
The linearized wavefront estimation problem in primary paper is:
\[
\text{minimize } \| \mathbf{GM}\mathbf{\phi} + \mathbf{g}_t \|_2^2 + \beta \| \nabla \mathbf{\phi} \|_2^2. \tag{S.1}
\]
recalling that matrix \( \mathbf{G} = \begin{bmatrix} \text{diag}(\mathbf{g}_x) & \text{diag}(\mathbf{g}_y) \end{bmatrix} \).

By introducing a slack variable \( \mathbf{w} \), whose physical interpretation is the wavefront gradient, the original objective function in Eq. (S.1) can be split into two parts, as:
\[
\text{minimize } \begin{aligned}
\| \mathbf{GMw} + \mathbf{g}_t \|_2^2 + 
\beta \| \nabla \mathbf{\phi} \|_2^2, 
\end{aligned} \tag{S.2}
\]
subject to \( \mathbf{w} = \nabla \mathbf{\phi} \).

Using ADMM then it yields Algorithm S.1, where \( \eta \) is the dual variable, and a “warm starting” numerical strategy is employed to fasten convergence, i.e. initializing the solution for one frame with the solution from previous frame. Now we briefly discuss how the \( \mathbf{\phi} \)-update step and \( \mathbf{w} \)-update step are computed.

Algorithm S.1 ADMM for solving Eq. (S.1).

1: procedure Reconstruct Wavefront(\( \mathbf{g}_x, \mathbf{g}_y, \mathbf{g}_t \))
2: Initialize \( \mathbf{\phi}^0, \mathbf{w}^0 \) and \( \eta^0 \) from previous frame, set \( \mu > 0 \);
3: while not converge do
4: \( \mathbf{\phi}^{k+1} \leftarrow \arg \min_{\mathbf{\phi}} f(\mathbf{\phi}) + \mu \| \nabla \mathbf{\phi} - \mathbf{w}^k + \eta^k \|_2^2; \)
5: \( \mathbf{w}^{k+1} \leftarrow \text{prox}_{\mu g}(\nabla \mathbf{\phi}^{k+1} + \eta^k); \)
6: \( \eta^{k+1} \leftarrow \eta^k + \nabla \mathbf{\phi}^{k+1} - \mathbf{w}^{k+1}; \)
7: end while
8: end procedure

\( \mathbf{\phi} \)-update step This involves solving a Poisson equation, which usually requires proper boundary conditions in conventional approaches, for example the periodic boundary condition [Hudgin 1977], or the Neumann boundary condition [Noll 1978]. However, in our case, because of the existence of \( \mathbf{M} \), the unknown boundary values are implicitly determined by minimizing the objective. When trivial boundary conditions are assumed, the solution to the resultant Poisson equation leads to non-trivial boundary values on the observed part of \( \mathbf{\phi} \). In practice, we found Neumann boundary condition suffices to yield a good estimation (see Figure S.1). Therefore, by just assuming Neumann boundary condition on the linear operators,

\[
\mathbf{\phi}^{k+1} = \arg \min_{\mathbf{\phi}} \beta \| \nabla \mathbf{\phi} \|_2^2 + \mu \| \nabla \mathbf{\phi} - \mathbf{w}^k + \eta^k \|_2^2 \]
\[
= \left( \beta/\mu + 1 \right) \nabla \mathbf{\phi}^{k+1} \]  
\[
= \mathcal{T}_{\text{DCT}}^{-1} \left( \mathcal{T}_{\text{DCT}} \left( \nabla \mathbf{\phi}^{k+1} - \eta^k \right) \right) \]  
\[
= \mathcal{T}_{\text{DCT}}^{-1} \left( \mathcal{T}_{\text{DCT}} \left( \nabla \mathbf{\phi}^{k+1} - \eta^k \right) \right). \tag{S.3}
\]

where the division is element-wise. Note that forward/inverse DCT can be efficiently implemented via forward/inverse Fast Fourier Transforms (FFT), respectively.

\( \mathbf{w} \)-update step This involves evaluation of \( \text{prox}_{\mu g}(\mathbf{u}) \), the proximal operator [Parikh et al. 2014] of \( g(\mathbf{w}) \) with parameter \( \mu \), which is defined and computed as:

\[
\text{prox}_{\mu g}(\mathbf{u}) = \arg \min_{\mathbf{w}} \| \mathbf{GMw} + \mathbf{g}_t \|_2^2 + \mu \| \mathbf{w} - \mathbf{u} \|_2^2 
\[
= \left( \mu I + \mathbf{M}^T \mathbf{G}^T \mathbf{G} \right)^{-1} \left( \mu \mathbf{w} - \mathbf{M}^T \mathbf{G}^T \mathbf{g}_t \right) 
\[
= \mathbf{M}^T \left( \mu I + \mathbf{G}^T \mathbf{G} \right)^{-1} \mu \mathbf{M} \mathbf{u} - \mathbf{M}^T \mathbf{G} \mathbf{g}_t + \left( \mathbf{I} - \mathbf{M}^T \mathbf{M} \right) \mathbf{u}. \tag{S.4}
\]

A closed form solution can be obtained for Eq. (S.4) because \( \mu I + \mathbf{G}^T \mathbf{G} \) is block-diagonal. Denote \( \mathbf{u} = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T \end{bmatrix}^T \), and use element-wise operator, then (notation: the arithmetic operators are element-wise):

\[
\text{prox}_{\mu g}(\mathbf{u}) = \begin{bmatrix} 
\mathbf{u}_1^T & \mathbf{u}_2^T \end{bmatrix}^T, \quad \text{if } \mathbf{u} \text{ outside boundary,}
\]
\[
\left[ \begin{array}{c}
\mathbf{g}_x^2 + \mu \mathbf{u}_1 & -\mathbf{g}_y^2 + \mu \mathbf{u}_1 \\
\mathbf{g}_y^2 + \mu \mathbf{u}_2 & -\mathbf{g}_x^2 + \mu \mathbf{u}_2 \\
\end{array} \right], \quad \text{otherwise.}
\]
Note that all the operations are either element-wise multiplications or divisions, and thus the computation of $\text{prox}_{g/\beta}(u)$ is highly efficient and is naturally parallelizable.

Comparison With Conjugate Gradient
In main text we compare our solver (Algorithm S.1) with conjugate gradient method that iteratively solves the unknown wavefront $\phi$ from the normal equation of Eq. (S.1):

$$\left(\nabla^T \mathbf{M}^T \mathbf{G}^T \mathbf{G} \nabla + \beta \nabla^2 \right) \phi = -\nabla^T \mathbf{M}^T \mathbf{G}^T g_t. \quad (S.5)$$

Note the linear system cannot be diagonalizable either in spatial or frequency domain. To fasten convergence, a reasonable approach would be to first solve an unconstrained flow estimation problem:

$$w^* = \arg \min_w \|G M w + g_t\|^2 + \beta \|w\|^2, \quad (S.6)$$

and then integrate the gradient $w^*$ to serve as an initial guess to the conjugate gradient solver, as $\phi^0 = (\nabla^2)^{-1}\nabla w^*$, assuming proper boundary condition. Using the previously introduced proximal operator notation, the solution can be written as $w^* = \text{prox}_{g/\beta}(0)$, with $g(w)$ defined previously and $0$ the null vector.

S.2 ADDITIONAL IMPLEMENTATION DETAILS
Timing
Figure S.2 shows that our GPU solver & controller runs in real-time with the camera sensor hardware synchronized to V-Sync, while the SLM is lacking the total speed because of its long response time. One whole AO iteration may probably take 5 or more V-Sync cycles, and hence reducing the total AO system performance to be only at around 10 Hz.

Fig. S.2. Example timing at one AO iteration (phase captured, reconstruction, and update).

Simulation
All parameters of the AO simulation in main text are listed in Table S.1. The generation of the turbulence wavefronts follows the subharmonics method [Lane et al. 1992], respecting Kolmogorov’s law [Kolmogorov 1941]. Deformation functionality is mimic by cubic interpolation.

Fabrication
The binary phase mask is fabricated on a 0.5 mm thick 4” Fused Silica wafer using photolithography techniques. We illustrate the fabrication pipeline in Figure S.3 and explain each step in details.

First, the designed binary phase (either 0 or $\pi$) is converted to a binary mask pattern (either 0 or 1 and written on a photomask by a laser direct writer Heidelberg DWL2000 (Figure S.3 (a)). Each pixel on the pattern is 12.9 $\mu$m. Second, the fused silica wafer is deposited with a 200 nm thick Cr film (Figure S.3 (b)) after cleaning in piranha solution. The Cr film will serve as a hard mask in the subsequent steps. Third, the fused silica wafer with Cr is spin-coated with a uniform layer of photore sist AZ1505 to form a 0.6 $\mu$m layer to be used in photolithography (Figure S.3 (c)). The photomask and the wafer coated with photoresist is then aligned on a contact aligner and photore sist is exposed by UV light. (e) The exposed areas on the photore sist are removed in the development. (f) Mask patterns are transferred to the Cr film by Cr etching. (g) The fused silica wafer is etching by Ar and SF$_6$ mixed plasma to obtain required depth. (h) After removing the residual Cr, the binary phase mask is finalized. (i) 3D profile of the central area on the fabricated binary phase mask taken with Zygo NewView 7300.

Fig. S.3. Fabrication of binary phase mask. (a) The designed mask patterns are first written on a Soda Lime photomask with laser direct writer. (b) A thin layer of Cr is deposited on one side of a fused silica wafer. (c) A uniform photore sist layer is formed on top of Cr by spin-coating. (d) The wafer and photomask is aligned on a contact aligner and photore sist is exposed by UV light. (e) The exposed areas on the photore sist are removed in the development. (f) Mask patterns are transferred to the Cr film by Cr etching. (g) The fused silica wafer is etching by Ar and SF$_6$ mixed plasma to obtain required depth. (h) After removing the residual Cr, the binary phase mask is finalized. (i) 3D profile of the central area on the fabricated binary phase mask taken with Zygo NewView 7300.
Table S.1. AO simulation parameters.

<table>
<thead>
<tr>
<th>Description</th>
<th>Shack-Hartmann AO</th>
<th>Curvature AO</th>
<th>High Resolution AO (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase Screen Pixel Size</td>
<td>1 µm</td>
<td>5 µm</td>
<td>5 µm</td>
</tr>
<tr>
<td>Phase Screen Resolution</td>
<td>5120 x 5120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor Resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrector Pixel Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Center Wavelength</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Propagation Distance z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bilateral Window Size</td>
<td>20 x 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Additional Parameters</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table S.2. Setup components.

<table>
<thead>
<tr>
<th>Description</th>
<th>Qty</th>
<th>Model</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plasma light source</td>
<td>1</td>
<td>HPLS245</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>Collimation adapter</td>
<td>1</td>
<td>cope3-a</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>Cage system ins</td>
<td>1</td>
<td>Cp208</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 longpass filter, cut-off 700 nm</td>
<td>1</td>
<td>FeldD00</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 achromatic doublets f = 200 mm</td>
<td>2</td>
<td>AC-254-200-A</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 achromatic doublets f = 150 mm</td>
<td>1</td>
<td>AC-254-150-A</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 wave-grad polarizer</td>
<td>1</td>
<td>WP25/M/115</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>2 linear polarizer</td>
<td>1</td>
<td>Lpvi5e30d-a</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>2 shortpass dichroic mirror</td>
<td>1</td>
<td>DmP650L</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>2 longpass dichroic mirror</td>
<td>1</td>
<td>DmP650L</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>2 beamsplitter cube</td>
<td>1</td>
<td>Rs031</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>XY linear translation stage</td>
<td>1</td>
<td>Xytum</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>Mounting stage for warped plates</td>
<td>1</td>
<td>Xy550M</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 long cage assembly rod</td>
<td>1</td>
<td>Eri</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>1 5 long cage assembly rod</td>
<td>1</td>
<td>Eri</td>
<td>Thorlabs</td>
</tr>
<tr>
<td>Reflective phase-only SLM</td>
<td>1</td>
<td>Plc-2-155-04-c</td>
<td>Holoeye</td>
</tr>
<tr>
<td>Camera body</td>
<td>1</td>
<td>Eos 70D</td>
<td>Canon</td>
</tr>
<tr>
<td>Camera macro lens</td>
<td>1</td>
<td>Ef 100 mm f/2.8</td>
<td>Canon</td>
</tr>
<tr>
<td>Waveform generator</td>
<td>1</td>
<td>1550mb2 Series</td>
<td>Keysight</td>
</tr>
<tr>
<td>Motorized stage</td>
<td>1</td>
<td>Ts5010/a</td>
<td>Zaber</td>
</tr>
<tr>
<td>8.6 mm x 8.6 mm square iris at SLM</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


Software Please refer to https://github.com/vccimaging/MegapixelAO for our open source repository.