Research Article Vol. 35, No. 6 / June 2018 / Journal of the Optical Society of America A 859

Journal of the Optical Society of America A

OPTICS, IMAGE SCIENCE, AND VISION

GPU implementation for spline-based wavefront reconstruction

Elisabeth Brunner,[1,*] Cornelis C. de Visser,[2] Cornelis Vuik,[3] and Michel Verhaegen[1]

[1]Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
[2]Department of Control & Simulation, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
[3]Delft Institute for Computational Science and Engineering, Delft University of Technology, Mekelweg 4, 2628 HS Delft, The Netherlands
*Corresponding author: a.e.brunner@tudelft.nl

Received 19 January 2018; revised 29 March 2018; accepted 30 March 2018; posted 3 April 2018 (Doc. ID 320064); published 2 May 2018

This paper presents an adaptation of the distributed-spline-based aberration reconstruction method for Shack–Hartmann (SH) slope measurements to extremely large-scale adaptive optics systems and the execution on graphics processing units (GPUs). The introduction of a hierarchical multi-level scheme for the elimination of piston offsets between the locally computed wavefront (WF) estimates solves the piston error propagation observed for a large number of partitions with the original version. To obtain a fully distributed method for WF correction, the projection of the phase estimates is locally approximated and applied in a distributed fashion, providing stable results for low and medium actuator coupling. An implementation of the method with the parallel computing platform CUDA exploits the inherently distributed nature of the algorithm. With a standard off-the-shelf GPU, the computation of the adaptive optics correction updates is accomplished in under 1 ms for the benchmark case of a 200 × 200 SH array.

OCIS codes: (010.1080) Active or adaptive optics; (010.7350) Wave-front sensing; (000.4430) Numerical approximation and analysis; (000.3860) Mathematical methods in physics; (010.1285) Atmospheric correction; (200.4960) Parallel processing.

https://doi.org/10.1364/JOSAA.35.000859

1. INTRODUCTION

To adequately compensate for the phase aberrations that are introduced by atmospheric turbulence, the new generation of extremely large-scale optical telescopes requires adaptive optics (AO) systems that scale with the size of the telescope pupil [1]. For the eXtreme-AO (XAO) system [2] of the future European Extremely Large Telescope (E-ELT), this places the number of wavefront (WF) sensor (WFS) measurements N in the range of 10^4–10^5. Because the actuator commands of the corrective device, consisting of a deformable mirror (DM), have to be updated at kilohertz-range frequencies, the work on fast algorithms to obtain estimates of the incoming WFs has been extensive [3–6], resulting in methods that reach linear computational complexity order [7–9]. However, for the dimensions of XAO systems, the boundaries in single CPU core performance pose a limit for methods that are based on inherently global solutions. Therefore, increasing efforts have been made in designing WF estimation algorithms specifically for parallel processing architectures.

The distributed-spline-based aberration reconstruction (D-SABRE) method [10] was recently introduced as an extension of the spline-based aberration reconstruction (SABRE) method [11]. The approach uses multivariate simplex B splines [12] to locally model WF aberrations and allows application on non-rectangular WFS arrays. In simulations for rectangular arrays, the WF estimates obtained with the SABRE method have proven superior to the classical finite difference method with Fried geometry [13] in terms of reconstruction accuracy and noise resilience [11]. The local nature of the B-spline basis functions can be exploited to derive an innately distributed solution to the WF reconstruction (WFR) problem. D-SABRE decomposes the global WFS domain into any number of partitions and computes the WF estimates in two distributed stages. In the first stage, local WFR problems, which are defined on the partitions and include only local WFS measurements, are solved in parallel, resulting in a set of local WF estimates. In the second stage, the distributed piston mode equalization (D-PME) equalizes the unknown piston modes of the local B-spline models to obtain the global WF estimate. The D-PME is an iterative process that requires only communication between neighboring partitions. The overall method has a theoretical computational complexity of $O(N^2/G^2)$ flops (floating point operations), which have to be performed per parallel processor, for a total of G partitions and scales, therefore linearly with the number of WFS measurements for $G \geq \sqrt{N}$ [10].

The D-SABRE method was extensively compared to the Cumulative Reconstructor with domain decomposition

(CuRe-D) method [14], a line integral approach with domain decomposition that has linear computational complexity and is suitable for parallel implementation. It has been observed that, constituting solutions to least-squares problems, the local D-SABRE WF estimates show good noise-rejection properties, whereas the cumulative approach of the CuRe-D algorithm leads to noise accumulation within the partitions. However, D-SABRE is subject to propagation of errors that are created in the estimation of the piston offsets between partitions, if a high level of domain composition is applied or if the domain contains large internal obscurations. Applying overlap between the partitions mitigated but did not negate this effect and also decreases the computational speed [10]. This phenomenon yields a trade-off in WF accuracy and number of partitions G, putting a limit on the latter, which prevents the D-SABRE method from reaching its full potential of linear or even sublinear computational load per processor for very or extremely large-scale AO systems.

The D-SABRE method in its current form was conceptualized for application with the most commonly used Shack–Hartmann (SH) WFS. The SH sensor is an array of lenslets that creates a focal spot pattern from which approximations of the local spatial WF gradients in each subaperture are derived [1,15]. Whereas the limited amount of processed data, i.e., two slope measurements per subaperture, restricts the slope-based D-SABRE method to linear B-spline polynomials, two approaches have been introduced that can extend the method to higher degree polynomials: (1) by exploiting the integrative nature of the SH sensor, a more advanced sensor model has been implemented that utilizes first- and second-order moment measurements of the focal spots [16]; and (2) by combining the standard D-SABRE with an additional correction step in which the pixel information in the focal spots is directly worked with, using an algorithm based on small aberration approximations of the focal spot models [17]. Employing a cubic B-spline representation of the phase, both approaches can achieve WF estimates that are superior to the linear D-SABRE WF estimates if applied to a given SH array. But because quadratic instead of linear sensor models are included, a trade-off with computation time has to be made.

This paper presents improvements to, and addresses remaining drawbacks of, the SH slope-based D-SABRE method to make it applicable to extremely large-scale AO systems while preserving the acquired strong points, i.e., locality and noise resilience of the analytical solution [10]. As a first contribution, we present an alternative approach for cancellation of the piston offsets between the local WF estimates. The hierarchical piston mode equalization (H-PME) is based on a multi-level approach that, rather than equalizing the piston mode in a partition local operation, levels tiles of partitions. Even though the H-PME requires communication not only between directly neighboring partitions but throughout the entire WFS domain, the necessary computations can be distributed. Whereas stricter requirements are posed on the shape of the triangulation to apply the H-PME, the procedure fixes the piston error propagation for the large number G of partitions and extends the applicability of the D-SABRE method to pupil shapes with arbitrarily large central obscurations [1]. To compensate for the present phase aberrations with an AO system, the DM actuator commands have to be computed such that the mirror shape optimally fits the estimated WF. Within the B-spline framework, we suggest an approach that performs this DM projection locally on each partition, which results, in combination with the D-SABRE method, in a fully distributed algorithm for fast updates of the corrective DM actuator commands. Developed as an inherently distributed algorithm, the D-SABRE method was intended for execution on parallel hardware. For the derivation of the per processor computational load mentioned earlier, all hardware-dependent issues, such as transport latency, cache size, and available instruction sets, were neglected [10]. The last contribution of this paper, an adaptation and implementation of the D-SABRE method for graphics processing units (GPUs), was therefore crucial to prove the potential of the approach to create scalability for the WF correction problem. Profiling results for the benchmark case of a 200×200 SH array showed that the presented GPU implementation reaches for XAO systems the required sub-millisecond computation times with off-the-shelf parallel hardware.

The organization of this paper is as follows. After a short introduction of the D-SABRE method and a description of the issues arising from the original PME procedure in Section 2, the alternative H-PME is presented in Section 3. Next to the formulation of the approach, numerical experiments with an end-to-end simulation tool for AO systems show the advantages in WFR accuracy and resilience to measurement noise. The procedure to compute the actuator command updates in a fully distributed manner and the approximation errors that are introduced compared to the global DM projection are described in Section 4. In Section 5, we discuss the GPU implementation of the D-SABRE method in detail and provide speed of the computations and memory transfers by timing. Finally, Section 6 concludes the paper.

2. PRELIMINARIES ON THE D-SABRE METHOD FOR WFR

The D-SABRE method consists of two stages, as illustrated in Fig. 1, with Stage 1 performing the distributed local WFR and Stage 2 the D-PME procedure [10]. An additional postsmoothing routine that was introduced as an optional addition to Stage 2 that not only eliminates the piston offsets but also imposes smoothness between the local estimates is not considered in this paper.

A. Stage 1: Distributed Local WFR

By constructing the global triangulation $\mathcal{T}$ on the reference centers of the SH subapertures, the B-spline model of the WF is defined in the pupil plane. In Fig. 2, the example of a regular Type-II triangulation, which will be used in the remainder of the paper, is depicted. The D-SABRE method is based on a decomposition of the global triangulation $\mathcal{T}$ into a set of G sub-triangulations such that

$$\mathcal{T} = \bigcup_{i=1}^{G} \mathcal{T}_i, \quad \mathcal{T}_i = \Omega_i \cup \Xi_i, \quad \Omega_i \cap \Xi_i = \emptyset. \tag{1}$$

Each sub-triangulation $\mathcal{T}_i$ consists of a core part Ω_i and an overlap part Ξ_i that, respectively, contain J_{Ω_i} and J_{Ξ_i} triangles,

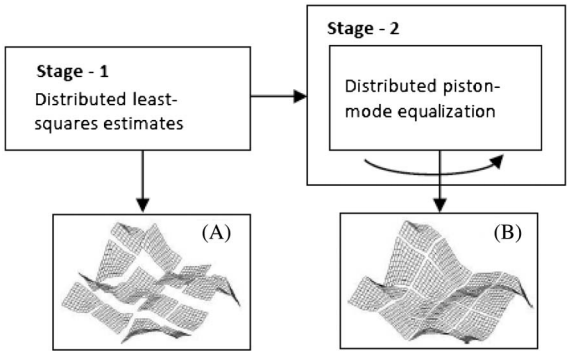

Fig. 1. Scheme of the D-SABRE algorithm: local WFR and D-PME.

resulting in $J_i = J_{\Omega_i} + J_{\Xi_i}$ triangles per partition. The width of the partition overlap is given in simplices and indicated with the *overlap level* (OL), with the example of Fig. 3 showing sub-triangulations with OL-1.

The local WF $\phi_i(x,y)$ within subpartition i is approximated through spline function $s_{r_i}^d(x,y)$ of polynomial degree d and continuity order r:

$$\phi_i(x,y) \approx s_{r_i}^d(x,y) = \mathbf{B}_i^d(x,y)\mathbf{c}_i, \qquad 1 \le i \le G, \tag{2}$$

where the local B-form matrix $\mathbf{B}_i^d(x,y)$ contains the B-spline basis functions and $\mathbf{c}_i \in \mathbb{R}^{J_i\hat{d}}$ the set of local B coefficients. The pupil plane coordinates are given by $(x,y) \in \mathbb{R}^2$, and $\hat{d} := \frac{(d+2)!}{2d!}$ denotes the total number of Bernstein polynomials per triangle. Note that for the SH slope-based D-SABRE, only linear splines of degree $d = 1$ with zero-order continuity can be employed, resulting in $\hat{d} = 3$.

The local slope vector $\sigma_i = [\sigma_{i,x}^{\mathrm{T}}, \sigma_{i,y}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2K_i \times 1}$ includes SH slope measurements of the subapertures for which the reference centers are located within sub-triangulation $\mathcal{T}_i$. The distributed local WFR amounts then to the following set of linear least-squares problems subjected to linear equality constraints:

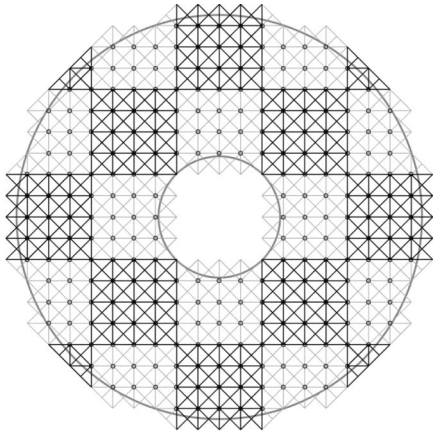

Fig. 2. D-SABRE Type-II triangulation [10] with 5 × 5 partitioning on a 16 × 16 SH array for a telescope with a central obscuration ratio of 0.3. The circles depict the reference centers of the SH subapertures that are illuminated with a minimum light ratio of 0.5 and the illumination area is outlined in gray.

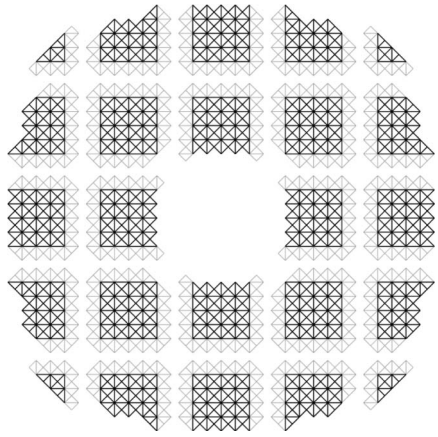

Fig. 3. D-SABRE sub-triangulations of the partitioning in Fig. 3 including partition overlap of OL-1 with core parts in black and overlap parts in gray.

$$\min_{\mathbf{c}_i \in \mathbb{R}^{J_i}} \|\sigma_i - \mathbf{D}_i\mathbf{c}_i\|_2^2 \quad \text{subject to } \mathbf{A}_i\mathbf{c}_i = 0, \qquad 1 \le i \le G, \tag{3}$$

where matrix $\mathbf{A}_i$ contains the continuity constraints that ensure smoothness of continuity order r within partition i. The local regression matrix is hereby defined as

$$\mathbf{D}_i := d\mathbf{B}_i^{d-1}(x,y)\mathbf{P}_{\mathbf{u}_i}^{d,d-1} \in \mathbb{R}^{2K_i \times J_i\hat{d}}, \tag{4}$$

with $\mathbf{B}_i^{d-1}(x,y)$ denoting the local B-form matrix for the reduced polynomial degree $d-1$ and $\mathbf{P}_{\mathbf{u}_i}^{d,d-1}$ the local de Casteljau matrix [18].

The local constraint matrix $\mathbf{A}_i$ is constructed from local smoothness matrix $\mathbf{H}_i$ and local anchor constraint $\mathbf{h}_i$:

$$\mathbf{A}_i := \begin{bmatrix} \mathbf{H}_i \\ \mathbf{h}_i \end{bmatrix} \in \mathbb{R}^{(R_i+1)\times J_i\hat{d}}, \tag{5}$$

where R_i is the number of local continuity constraints in sub-triangulation $\mathcal{T}_i$. The anchor vector $\mathbf{h}_i = [1 \quad 0 \quad \cdots \quad 0] \in \mathbb{R}^{1\times J_i\hat{d}}$ fixes the local piston modes to a predefined constant.

The local WFR problems, each consisting of a linear least-squares problem with equality constraints, is solved through projection onto the null spaces of the local constraint matrices:

$$\overline{\mathbf{c}}_i = (\overline{\mathbf{D}}_\mathbf{i}\overline{\mathbf{D}}_\mathbf{i}^{\mathrm{T}})^{-1}\overline{\mathbf{D}}_\mathbf{i}^{\mathrm{T}}\sigma_i, \tag{6}$$

where the projected local regression matrix $\overline{\mathbf{D}}_\mathbf{i} := \mathbf{D}_\mathbf{i}\mathbf{N}_{\mathbf{A}_i}$ is obtained with an orthogonal basis of the null space of $\mathbf{A}_i$ stored in matrix $\mathbf{N}_{\mathbf{A}_i} := \mathrm{null}(\mathbf{A}_i) \in \mathbb{R}^{J_i\hat{d}\times\overline{d}_i}$, where $\overline{d}_i < J_i\hat{d}$. The result is still in the null space of the constraint matrix, and the final local coefficient vector is retrieved by evaluating the vector space of $\mathbf{N}_{\mathbf{A}_i}$ with

$$\mathbf{c}_i^* = \mathbf{N}_{\mathbf{A}_i}\overline{\mathbf{c}}_i. \tag{7}$$

B. Stage 2: D-PME

After computation of the local WF estimates in Stage 1, the unknown local piston modes have to be equalized in Stage 2, for which the D-PME was introduced [10].

Let $\mathcal{M}_i$ be the index set of all neighbor partitions m to partition i. The index vector $\Omega_{i,m}$ collects the coefficients within the core part of partition i that are located on the border shared

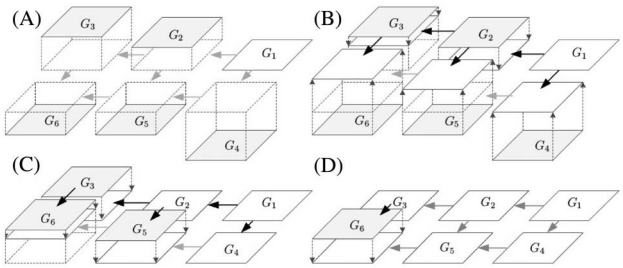

Fig. 4. D-PME operation on a 2×3 partitioning with the to-be-leveled sub-triangulations labeled as G_i, for $i = 1, \cdots, 6$.

with the core part of neighbor partition m, and $\Omega_{m,i}$ does so vice versa. With the results of Stage 1 from Eq. (7) stacked in the global starting vector $\mathbf{c}(0)$ of the D-PME, iteration l of the procedure is then formulated as

$$k_i(l+1) = \max_m \{|\mathbf{c}(l)_{\Omega_{i,m}} - \mathbf{c}(l)_{\Omega_{m,i}}|\}, \quad (m > i) \in \mathcal{M}_i, \tag{8}$$

$$\mathbf{c}_{\Omega_i}(l+1) = \mathbf{c}_{\Omega_i}(l) + k_i(l+1), \tag{9}$$

where $|\cdot|$ stands for the mean of the coefficient difference vectors, and Ω_i denotes as subscript the coefficients located within the core part of the sub-triangulations. The D-PME operation is illustrated for a small-scale example in Fig. 4.

The computation of the piston offset and the update of the local coefficient vector in Eqs. (8) and (9) require only access to the coefficients of the local partition and its neighbors and can therefore be performed in a distributed fashion.

1. Chain Propagation in the D-PME

The asymmetry of using the *maximum* offset between partition i and only neighbor partitions m with $m > i$ is necessary for the D-PME operation to converge. It causes the information flow to propagate sequentially through the partitioning, during which the partitions keep adapting their piston modes in a distributed fashion until they are all equalized relative to one predefined partition, the *master partition*. The number of iterations L_D in which the D-PME converges is hence given by the maximum distance, counted in partitions, between the master partition and any other partitions, resulting in a minimum of $L_D = \frac{\sqrt{G}}{2}$ iterations.

Although crucial for convergence of the method, it has been shown in numerical experiments that there are major drawbacks to the asymmetric equalization of the piston offsets.

Inaccuracies in the computation of the piston offsets between partitions, caused by errors in the local WF estimates provided by Stage 1, are propagated along *partition chains* throughout the grid of sub-triangulations. This effect is magnified by stronger decomposition of a given triangulation, in particular in the presence of large amounts of measurement noise. The decreased size of the sub-triangulations deteriorates the accuracy of the piston offset estimations, because a smaller number of coefficients, shared by the considered neighboring partitions, are used to compute the offsets in Eq. (8). The increased number of sub-triangulations in the partitioning aggravates the accumulation of the resulting PME errors, which yields to partition chains that diverge in terms of their piston modes. Application of large amounts of partition overlap can mitigate but not resolve the problem and also reduce the computational speed of the D-SABRE method [10].

Increasing the partition overlap does not suffice to restore a satisfying performance of the D-PME algorithm, if the D-SABRE is applied to a telescope pupil with a central obscuration that is large enough to interrupt the partition chains along which the PME information propagates. An example is shown in Fig. 2, where the decomposition of the global triangulation that is built on the illuminated subapertures of the SH array results in neighboring partitions that do not share any coefficients. In this case, the asymmetry of the D-PME operation does not allow an information flow around the obscuration but causes large PME errors that are then further propagated along the partition chains. The authors of this paper see the potential to render the D-PME procedure applicable to such cases by formulating the D-PME as a consensus enforcing distributed optimization problem, as it can be, e.g., realized with the alternating direction method of multipliers (ADMM) [19]. In this paper, however, a different procedure based on a multi-level approach was realized, where information exchange occurs between groups of partitions instead of merely directly neighboring partitions.

3. H-PME

We present an alternative PME procedure that resolves the issues of chain and PME error propagation: the H-PME. Like the D-PME, the H-PME procedure is designed as a distributed algorithm; however, it realizes the information flow not in a sequential but a hierarchical manner.

A. H-PME Procedure

Whereas the D-PME can be applied to non-rectangular partitionings, the H-PME requires a more rigid decomposition of the global triangulation into a square $2^p \times 2^p$ grid of G sub-triangulations $\mathcal{T}_i$, for $p \in \mathbb{N}$, resulting in $G = 2^{2p}$ partitions.

The H-PME method is performed in several levels $h = 1, \cdots, p$. In each level h, square sub-grids of the partitioning are grouped into *partition tiles*, each containing $2^{2(h-1)}$ neighboring partitions, as is visualized in Fig. 5. The resulting grid of partition tiles is then organized in $2^{2(p-h)}$ groups that

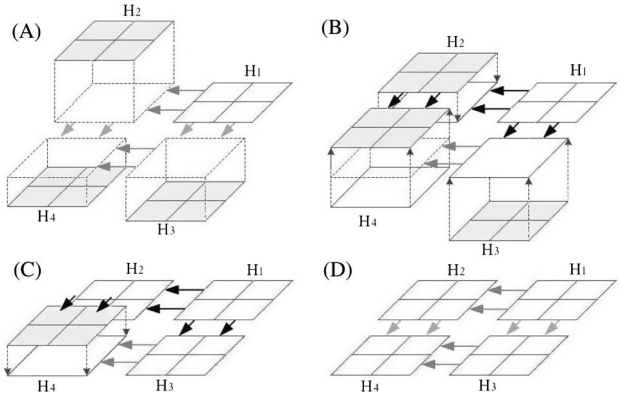

Fig. 5. Level 2 of the H-PME operation on a 4×4 partitioning, equalizing tiles H_t, $t = 1, \cdots, 4$ each containing four partitions.

each contain four neighboring tiles. The PME of level h is performed within these groups. Figure 5 depicts the second and last level of the H-PME on a 4×4 partitioning, which acts on one group of tiles each containing four partitions. The tiles, labeled as H_2, H_3 and H_4, are leveled in three steps with respect to tile H_1.

The H-PME is formulated as the following procedure. For the levels $h = 1, \cdots, p$, let $\mathcal{H}^h_{g,t}$ be the index set of all partitions i that are contained in the partition-tile t of group g, where $t = 1, \cdots, 4$ and $g = 1, \cdots, 2^{2(p-h)}$. To equalize the piston offsets in group g, three offsets k_{g,t_i} have to be computed that level three tiles with respect to a master tile in the group. For these three offsets, the target tile is indexed as t_i and the master tile as t_m according to a predefined order. With the global starting vector $\mathbf{c}(0)$ obtained from Stage 1, the piston correction of tile t_i in group g at level h of the H-PME is given as

$$k_{g,t_i}(h) = \frac{1}{2^{h-1}} \sum_{i,m} |\mathbf{c}(h-1)_{\Omega_{i,m}} - \mathbf{c}(h-1)_{\Omega_{m,i}}|,$$
$$\text{for all } i \in \mathcal{H}^h_{g,t_i}, m \in \mathcal{H}^h_{g,t_m}$$
$$\text{with } \Omega_{i,m} \cap \Omega_{m,i} \neq \emptyset. \tag{10}$$

$$\mathbf{c}_{\Omega_i}(h) = \mathbf{c}_{\Omega_i}(h-1) + k_{g,t_i}(h), \text{ for all } i \in \mathcal{H}^h_{g,t_i}. \tag{11}$$

Hereby, the piston offset is applied to all partitions in the target tile t_i of the considered group g. It is computed as the mean of all differences between coefficients that are located at the shared partition borders of the considered tiles. As in the D-PME procedure, $\mathbf{\Omega}_{i,m}$ is the index vector to all coefficients in partition i that are shared with partition m and vice versa. Which target tiles within the group of four are leveled to which master tile is predefined by the setting of t_i and t_m. In Fig. 5, the three tile offsets were performed according to the order $t_i = 2, 3, 4$ and $t_m = 1, 1, 2$.

As for the D-PME, the update rule of the H-PME procedure in Eq. (11) can be performed distributedly for all partitions. The computations of the piston offsets k_{g,t_i} in Eq. (10), however, are not a partition local (including only direct neighbor partitions) operation, as this is the case for the D-PME. The higher the considered level of the H-PME procedure, the more spread out the required information is through the global triangulation. Because there is no intersection between the partition index sets $\mathcal{H}^h_{g,t}$ of two different groups g_1 and g_2, parts of the computation can nevertheless be performed in a distributed manner.

It has already been stated that the H-PME is restricted to square $2^p \times 2^p$, $p \in \mathbb{N}$, partitionings for which the procedure is performed in p levels. In terms of the total number of partitions G, this leads to $L_H = \log_2(\sqrt{G})$ iterations for the H-PME, resulting in a faster information flow with the hierarchical scheme than with the sequential scheme of Section 2.B.

The following section demonstrates in numerical experiments that the H-PME does not suffer from the increased PME error propagation for strong decompositions of the triangulation that was discussed in Section 2.B. It is further shown that adequate PME is also possible in the presence of a central obscuration when the H-PME is applied.

B. H-PME in Numerical Experiments

To compare the D- and H-PME procedures, the D-SABRE open-loop WFR accuracy achieved under the presence of measurement noise is tested for both PME procedures. The Object–Oriented MATLAB Adaptive Optics (OOMAO) simulation tool [20] was used to numerically generate an AO system with an on-axis natural guide star. The experiments consist of Monte Carlo simulations that are based on 100 WF realizations computed for atmospheric turbulence of 15 cm Fried parameter at wavelength 550 nm. The D-SABRE method is applied to sets of the diffraction-based SH slope measurements that are obtained from the simulation tool. The B-spline estimate of the WF is evaluated at the resolution of the simulated phase screens, and the reconstruction accuracy is given in relative RMS (root mean square) error, i.e., the ratio between the residual and the aberration RMS, and averaged over all realizations.

1. Elimination of the Piston Error Propagation

The first experiment purely investigates the effect of the number of partitions on the reconstruction accuracy for a given SH array size and therefore considers a square telescope pupil, because non-rectangular pupils influence the performance of the PME procedures. The D-SABRE method is applied for different decomposition levels employing both the D- and the H-PME scheme. A 64×64 SH lenslet array that is fully illuminated by the square pupil of side length 25 m is simulated. The slope measurements are exposed to increasing levels of photon-shot noise that are indicated by the decreasing guide-star magnitude. The D-SABRE runs on a regular Type-II triangulation [10], and the local WFRs are obtained with a minimal partition overlap of OL-1. In the results presented in Fig. 6, the problematic piston mode error propagation of the D-PME procedure can be observed for the highly

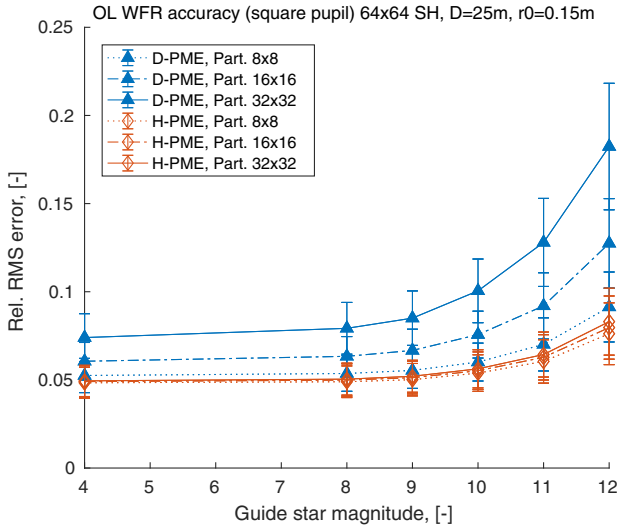

Fig. 6. Reconstruction accuracy and noise resilience comparing D-PME and H-PME for different levels of decomposition and increasing presence of photon shot noise. A square pupil was assumed; hence all subapertures in the considered 64×64 SH array are illuminated. Part., partitioning; Rel., relative.

decomposed 16 × 16 and 32 × 32 partitionings at all levels of photon-shot noise. The reconstruction accuracy obtained with the H-PME is hardly affected by the increase in the number of partitions and the reduction of the partition sizes. The hierarchical procedure is also less affected by the increase in photon-shot noise. Unlike the D-PME, in which all piston offsets are sampled locally based only on coefficients of a single partition, the H-PME computes the piston offset in higher levels as averages of coefficient differences obtained from several neighboring partitions. This averaging of partition offsets not only prevents propagation of local PME errors but also improves the noise-rejection properties of the procedure.

2. Inclusion of an Annular Pupil through Zero Padding

Second, the applicability of the H-PME procedure to telescope pupils with central obscuration is shown. An annular telescope pupil with a diameter of 25 m and 0.3 obscuration ratio is considered. From the 64 × 64 SH array, the slopes measurements of subapertures with an illumination of at least 50% are processed. Due to the requirement of a square $2^p \times 2^p$ partitioning, $p \in \mathbb{N}$, it is not possible to construct the global triangulation on the reference centers of only the illuminated subapertures, as seen in Fig. 2. Strong decomposition of such annular triangulations will create sub-triangulations to which no data are assigned and hence yield non-square partition grids, as discussed in Section 2.B. To allow arbitrary pupil shapes and SH array dimensions, the illuminated subapertures are embedded into a square SH array that returns zero-slope measurements. Based on this *zero padding* of the SH data, an appropriate triangulation and partitioning can be created (see Figs. 7 and 8). At each level of the H-PME, the orientation of the partition tile leveling of Eqs. (10) and (11) is performed such that tiles with the largest number of coefficients within the pupil are considered first. This scheme has also been tested for D-PME by considering only coefficients located within the pupil for Eqs. (8) and (9). However, even though a slight improvement was observed, the piston error chains could not be sufficiently reduced to obtain useful results, as can be seen in the top phase screens of Fig. 9, which were computed in a noise-free scenario. Whereas the residual phase obtained with the H-PME (bottom of Fig. 9) also shows remaining piston offsets for several partitions at the edge of the pupil, these remain local. Furthermore, an overall equalization of the piston modes is achieved.

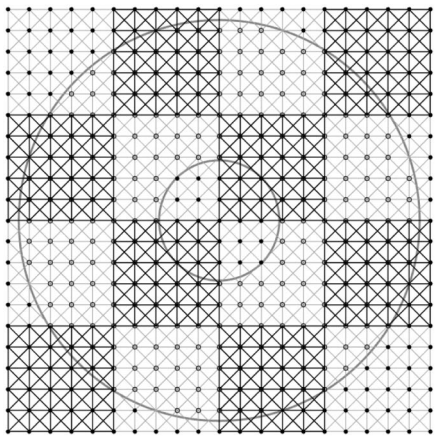

Fig. 7. Zero-padded D-SABRE Type-II triangulation with 4 × 4 partitioning on a 16 × 16 SH array for a telescope with a central obscuration ratio of 0.3. The circles depict the reference centers of the SH subapertures that are illuminated with a minimum light ratio of 0.5, and the illumination area is outlined in gray.

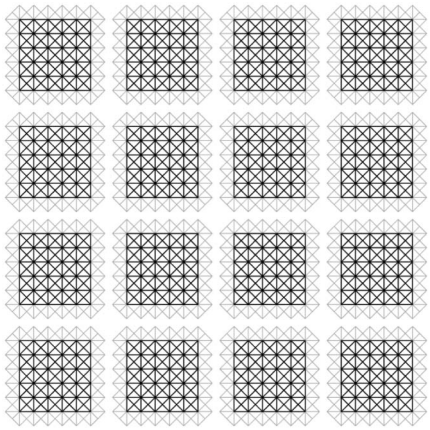

Fig. 8. Zero-padded D-SABRE sub-triangulations of the partitioning in Fig. 7, including partition overlap of OL-1 with core parts in black and overlap parts in gray.

The results of a Monte Carlo simulation under influence of measurement noise, obtained for the described setup of a round pupil with central obscuration, are presented in Fig. 10. In addition to increasing levels of photon-shot noise, the SH array is further exposed to a constant level of two electrons readout noise per pixel [20]. After embedding of the illuminated SH lenslets of the system into a square array, the sub-triangulations are created for several decomposition levels. After local reconstruction that includes both zero and actual slopes, the D-PME and H-PME are applied using only coefficients located within the illuminated area in which also the residual WF is computed. As expected, the D-PME provides reconstruction accuracy that is much inferior to the results obtained with H-PME. The massive jump in RMS error observed for the D-PME when applying stronger decomposition shows that entire partitions in the central obscuration are without real data that interrupt the partition chains along which the information propagates. Affected by errors that are introduced at the edges of the pupil through the zero data, the H-PME loses performance if compared with the square pupil experiment of Fig. 6. An interesting observation that can be made is that stronger decomposition of the triangulation improves the result, because the reduced partition sizes contain the erroneous piston estimates in a smaller area of the pupil. Also, it should be mentioned that the effect of the zero padding will be less strong in a closed-loop (CL) scenario where the real slopes are smaller.

Nevertheless, efforts should be undertaken to reduce the effects of the zero data visible at the edges of the illuminated part of the telescope pupil, particularly because on a real site the telescope spiders [2] supporting the secondary mirror create additional obscured areas. Extrapolating the slope data to avoid sharp features at the edges of the pupil and the central obscuration would be an option to consider. Also extending the

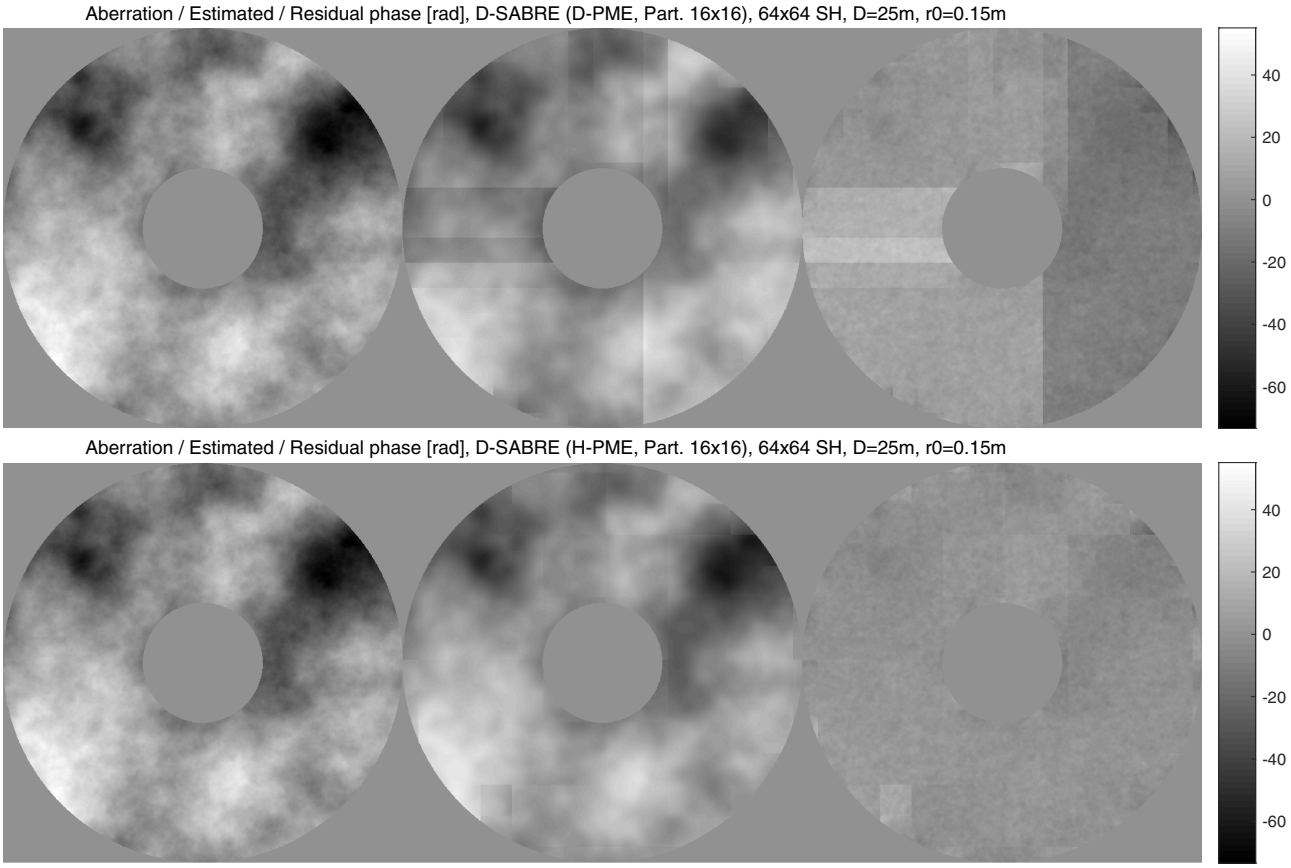

Fig. 9. Phase screens of aberration, estimate, and residual computed for an annular pupil with central obscuration of ratio 30% with the D-SABRE method applying the D-PME (top) and the H-PME (bottom) procedures. Piston mode errors caused by zero slopes that are processed for the non-illuminated subapertures of the 64 × 64 SH array can be observed. Part., partitioning.

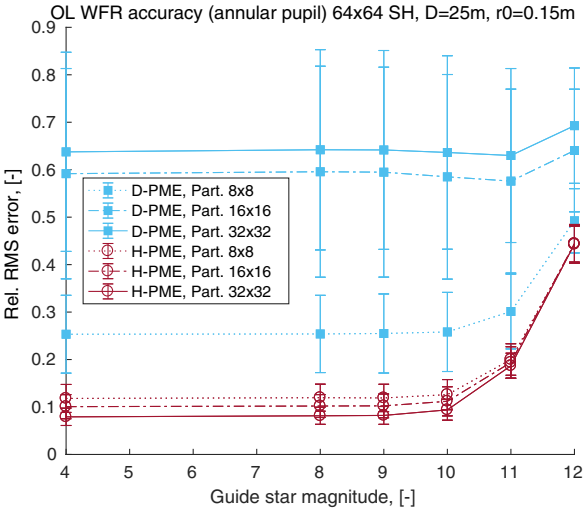

Fig. 10. Reconstruction accuracy and noise resilience comparing D-PME and H-PME on a zero-padded triangulation (subapertures within pupil illuminated, zero data processed outside of pupil) for different levels of decomposition and increasing presence of photon shot noise. The estimates were computed on the full 64 × 64 SH array and compared within the annular pupil (central obscuration of 30%). Part., partitioning; Rel., relative.

H-PME procedures to non-square partition grids and reducing the resolution of the triangulation to bridge the spider obscurations should be part of further studies aiming at avoiding zero padding altogether. Another issue not in the scope of this work is the consideration of differential piston effects due to pupil segmentation, which results from segmented mirrors [21] and wide spiders.

4. DISTRIBUTED DM PROJECTION

Once the WF estimate is retrieved with the D-SABRE method, a local solution for the projection onto the space of actuator commands driving the DM has to be investigated to obtain WF correction in a fully distributed manner.

A. Distributed Projection Problem

For a given actuator command vector $\mathbf{u} \in \mathbb{R}^M$, the global phase $\phi_{\mathbf{u}}$ introduced by the DM is represented as

$$\phi_{\mathbf{u}}(x, y) = \mathbf{F}(x, y)\mathbf{u}, \tag{12}$$

where matrix $\mathbf{F}(x, y)$ contains the values of the actuator influence functions at the pupil domain coordinates $(x, y) \in \mathbb{R}^2$. The structure and sparsity of influence matrix $\mathbf{F}(x, y)$ depends on the shape of the influence functions and the placement of the actuators and their intercoupling behavior.

The global DM projection has to find the set of actuator commands that optimally fit the DM phase $\phi_{\mathbf{u}}(x,y)$ to the estimated global D-SABRE phase estimate $\phi(x,y) = \mathbf{B}^d(x,y)\mathbf{c}$, at all pupil plane locations (x,y) within the telescope aperture. This can be achieved by solving the following least-squares problem:

$$\min_{\mathbf{u}\in\mathbb{R}^M} \|\mathbf{B}^d(x,y)\mathbf{c} - \mathbf{F}(x,y)\mathbf{u}\|_2^2, \tag{13}$$

where $\mathbf{B}^d(x,y)$ is the global B-form matrix and $\mathbf{c} \in \mathbb{R}^{J\hat{d}}$ the global B-coefficient vector obtained from the D-SABRE method.

For the distributed DM projection, a local actuator fitting problem is constructed for each partition i where only the local phase estimates represented by the local coefficient vector $\mathbf{c}_i \in \mathbb{R}^{J_i\hat{d}}$ and the commands $\mathbf{u}_i \in \mathbb{R}^{M_i}$ of actuators located within the respective partition are considered. Because the actuators are subjected to intercoupling, the B coefficients and actuators located in the entire sub-triangulation $\mathcal{T}_i$, including core and overlap parts Ω_i and Ξ_i, are matched to mitigate the effect. Note that to do so, the coefficient offsets within the PME procedures have to be applied not only to the partition core but to the partition overlap as well.

The local DM projection problem of partition i is then formulated as the least-squares problem

$$\min_{\mathbf{u}_i\in\mathbb{R}^{M_i}} \|\mathbf{B}_i(x,y)\mathbf{c}_i - \mathbf{F}_i(x,y)\mathbf{u}_i\|_2^2, \tag{14}$$

for local B-form matrix $\mathbf{B}_i(x,y)$ that is evaluated at pupil plane coordinates (x,y) within sub-triangulation $\mathcal{T}_i$. The values of the influence functions for the actuators located within $\mathcal{T}_i$ are sampled at the same coordinates and collected in matrix $\mathbf{F}_i(x,y)$. Because the local influence matrices are constructed for overlapping parts of the pupil plane coordinate plane and actuator grid, $\mathbf{F}_i(x,y)$ cannot simply be retrieved as blocks of the global influence matrix $\mathbf{F}(x,y)$. With the resulting optimal local actuator commands, given by

$$\mathbf{u}_i = (\mathbf{F}_i(x,y)^{\mathrm{T}}\mathbf{F}_i(x,y))^{-1}\mathbf{F}_i(x,y)^{\mathrm{T}}\mathbf{B}_i(x,y)\mathbf{c}_i, \tag{15}$$

the global actuator command vector $\mathbf{u}$ can be constructed from the commands $\mathbf{u}_{\Omega_i}$ of the actuators located within the core parts Ω_i of each partition.

It is important to mention that this approach is a very simple and minimalistic solution to the distributed DM projection, which is expected to introduce errors for very strong actuator coupling. To obtain the solution of the global DM projection problem of Eq. (13) in a distributed manner without approximation errors, future work should investigate the formulation of the distributed DM projection problem, e.g., as a sharing optimization problem with ADMM [19]. Coupling constraints on local command vectors $\mathbf{u}_i$ can be employed to achieve consensus between actuators that are shared by neighboring partitions or whose influence functions reach neighboring partitions [17].

To understand the range of applicability for the simpler presented distributed DM projection, numerical experiments for AO systems in a CL scenario, obtained with the simulation tool OOMAO, are shown in the next section.

B. Distributed DM Projection in Numerical Experiments

This section investigates the effects of not only performing the WFR but also the projection onto the DM in a distributed manner. Another AO system with an on-axis natural guide star, this time in a CL setting, was simulated with OOMAO. To ease the computational load of running the Monte Carlo simulations, a smaller system with a 32×32 SH array and a telescope with 12 m diameter are considered. The numerical DM is created with a built-in set of modes, derived from cubic Bezier curves, that result in a local region of influence of the actuators on the DM phase [20]. A Fried geometry was chosen, locating the actuators in the pupil plane on the corner of the subapertures and yielding a 33×33 actuator array if the entire SH array is considered.

1. Influence of Actuator Coupling

As a first experiment, the WF correction obtained with the D-SABRE (with H-PME) for a fully illuminated SH array, i.e., a square telescope pupil, is investigated to not take into account the effects of the zero padding (see Section 3.B) but to purely compare the impact of using the distributed DM projection instead of the global projection for varying levels of actuator coupling and different loop gains. A low-level noise scenario, with a natural guide star of magnitude 8 and two electrons readout noise per pixel, was adapted.

As a performance measure, the long-exposure Strehl ratio [1] (ranging from 0 to optimally 1) was computed from a simulated science camera that pictures an on-axis science star created in J band. The clock rate of the camera is with 500 Hz set equal to the sampling time of the telescopes, and integration lasts for an exposure of 500 frames starting after the first 20 frames [20]. To obtain a statistic for the experiment, it was performed for 50 sets of phase screens propagating at a wind speed of 10 km/h.

The plots in Fig. 11 show the Strehl ratios for WF corrections based on D-SABRE WF estimates that are projected onto the DM with either the global solution of Eq. (13) or the distributed solution of Eq. (14). Both versions were run for a moderate 4×4 and a very strong 16×16 decomposition of the triangulation and are compared to the baseline result obtained with the global SABRE WF correction. The WF correction performance was tested for different amounts of actuator couplings. Low couplings of 20% and 30% see inferior Strehl ratios than the strong couplings of 40% and 50% throughout, due to inadequate actuator spacing for the considered Fried parameter of $r_0 = 15$ cm [22]. This also leads to large variances in the low-coupling results, which are magnified by a small number of outliers occurring for all tested versions. Of main interest, however, is the behavior of the methods relative to each other. The D-SABRE WFR with global DM projection provides slightly lower but, relative to the global SABRE correction, fairly constant correction quality for all considered actuator couplings and loop gains. The D-SABRE with distributed DM projection is sensitive to both parameters. In case of both low coupling and gain, it provides converging correction for all considered loop gains and even, by a narrow margin, outperforms the D-SABRE with global DM projection. However, for stronger couplings, errors introduced by the

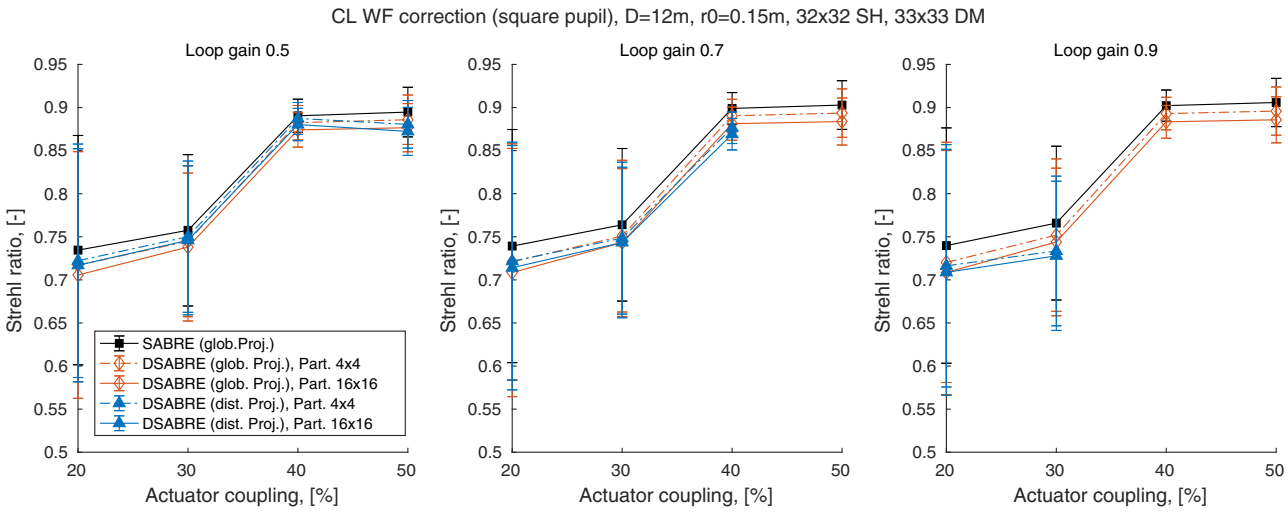

Fig. 11. Correction performance in long-exposure Strehl ratio comparing SABRE and D-SABRE with global and distributed DM projection for different levels of decomposition. In a noise-free scenario, different levels of actuator coupling and three levels of loop gain and a fully illuminated 32 × 32 SH array are considered. dist. Proj., distributed projection; glob. Proj., global projection; Part., partitioning.

approximation of the global DM projection problem in Eq. (13) with the local DM projection problems in Eq. (14) show effect. Divergence of the correction obtained with the D-SABRE and distributed DM projection for actuator couplings of 40% and 50% could only be prevented by lowering the loop gain to 0.7 and 0.5, respectively. For all coupling and gain combinations, it can further be observed that very strong partitionings, resulting in very small sub-triangulations only covering a few subapertures, yield a drop of about 1% in the achieved Strehl ratio for both D-SABRE corrections.

2. Application on an Annular Pupil

In a second experiment, the D-SABRE with global and distributed DM projection is applied to an annular telescope pupil with central obscuration. As in Section 3.B, the SH slopes are embedded in a square array of zero-slope measurements for the D-SABRE runs. The SABRE with global projection, which does not require any zero padding and is performed on the illuminated part of the sensor domain, is given as a baseline result in Fig. 12. The experiment records the long-exposure Strehl ratio for presence of increasing amounts of photon shot noise and, again, a constant level of two electrons of readout noise per pixel.

At the end of Section 3.B, we suggested that the piston errors, which remain in the open-loop estimates after the H-PME procedure (see Fig. 9) and are effects of the zero padding, would be reduced in CL. Indeed, there are no visible remnants of the piston mode errors in the corrective and residual phase screens of Fig. 13, which were retrieved after 20 CL iterations of D-SABRE with distributed DM projection in a noise-free setting. Even though measurement noise is expected to increase the impact of zero padding, this observation gives a positive outlook.

With an actuator coupling of 40% and a loop gain of 0.7, the D-SABRE with distributed DM projection is now tested for noise resilience in a setting for which the method found itself close to the limits of its applicability in the previous experiment (see Fig. 11). The results in Fig. 12 show that for guide start magnitudes $\lesssim 8$, the fully distributed procedure achieves Strehl ratios within 0.8% of the global SABRE baseline for the moderated 4 × 4 partitioning and within 1.4% for the very strong 16 × 16 partitioning. The deterioration of the distributedly computed correction increases for stronger noise levels.

The most striking finding in this experiment is, however, the fact that for illumination through an annular pupil, the distributed DM projection outperforms the global alternative when reconstructing the WF with the D-SABRE. This can be explained with local WFR and piston mode estimation errors

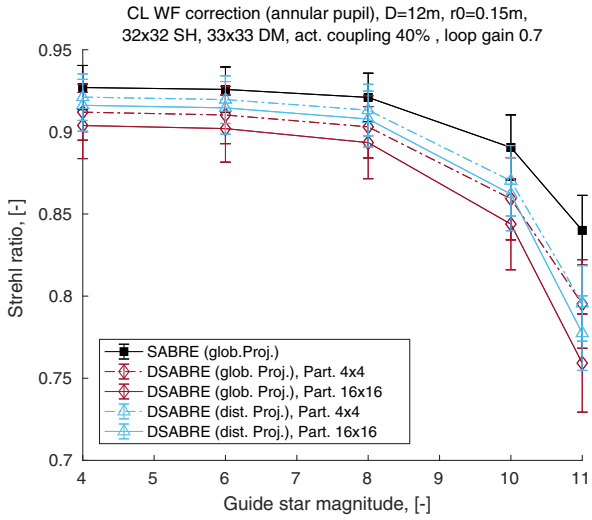

Fig. 12. Correction performance and noise resilience in long-exposure Strehl ratio comparing SABRE and D-SABRE with global and distributed DM projection for two levels of decomposition and increasing presence of photon shot noise. The estimates were computed on the full, zero-padded 32 × 32 SH array and the correction applied within the annular pupil (30% central obscuration). act., actuator; dist. Proj., distributed projection; glob. Proj., global projection; Part., partitioning.

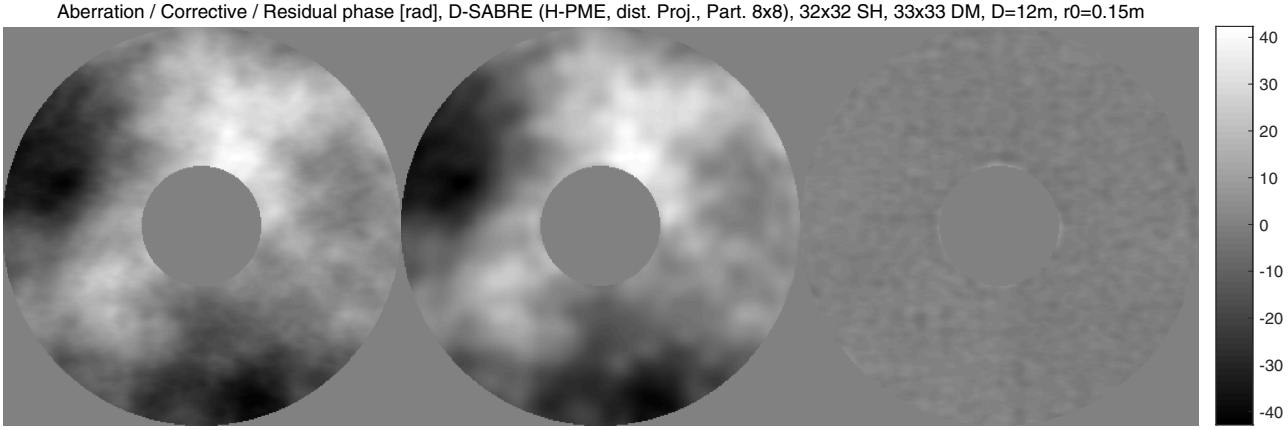

Fig. 13. Phase screens of aberration, correction, and residual after 20 CL iterations using a loop gain of 0.7. The D-SABRE method with the H-PME procedure and the distributed DM projection was applied to a 32 × 32 SH array, a 33 × 33 DM (40% actuator coupling), and an annular pupil with central obscuration of ratio 30%. There are no visible remnants of the piston mode errors caused by zero slopes that were observed in the open-loop setting (see Fig. 9). dist. Proj., distributed projection.

introduced through measurement noise and zero padding of the non-illuminated subapertures. Whereas these errors remain local with the distributed projection, a propagation throughout the telescope pupil takes place with the global projection.

5. ADAPTATION OF D-SABRE FOR GPUS

Currently, GPUs are popular for many engineering applications [23,24]. With a high computational load in the fully distributed parts of the method due to full local reconstruction and projection matrices, and with low interpartition communication for the PME procedures, the D-SABRE method was conceptualized for implementation on parallel hardware, specifically on GPUs.

GPUs are specialized for compute-intensive, highly parallel computation. A GPU is built around an array of streaming multiprocessors (SMs), with a certain number of GPU cores allocated to each SM. The implementation of the D-SABRE method presented in this chapter is programmed with CUDA, a parallel computing platform and programming model provided by NVIDIA [25]. CUDA enables the definition of C functions, called *kernels*, that are executed in parallel by CUDA *threads* on single-processor cores. Threads are then grouped into so-called *thread blocks* that execute independently from each other on different SMs, creating scalability. All threads in a thread block have access to some *shared memory* on the respective SM, which allows cooperative but parallel computation within a block. This *fine-grained data parallelism* is embedded within *coarse-grained data parallelism* among the thread blocks. The number of threads per block is, on current GPUs, limited to 1024 [25].

The efforts undertaken to optimize the performance of the CUDA program, which executes the D-SABRE on the GPU, presented in this paper can be summarized as follows: maximizing utilization by optimally exploiting parallelism in the algorithm and translating it to the hardware; minimizing data transfers with low bandwidth, with a focus on minimal data transfer between CPU (host) and GPU (device); and increasing instruction throughput by the use of single-precision floating point numbers and, if possible, avoidance of synchronization points.

A. Distributed WF Reconstruction and DM Projection as Matrix-Matrix Product

The computationally most expensive operations in the D-SABRE method are the (full) matrix-vector (MV) products of the local WF reconstructions and, in a CL setting, DM projections in Eqs. (6) and (14).

Operations that allow a high computational load per thread at a low required memory transfer, referred to as *compute-bound* rather than *memory-bound* problems, are most favorable for execution on a GPU. A prime example of such an operation is the matrix-matrix (MM) product. Reformulating the local reconstructions and projections to a single multiplication of two matrices has proven to be key in pushing the computation time to the required sub-millisecond range for extremely large sensor and actuator arrays of $N = \mathcal{O}(10^4)$.

To realize the distributed operations as a single MM product, all partitions $i = 1, \cdots, G$ need to carry identical reconstruction and projections matrices. The local input and output vectors are then of the same size and can be stacked in matrices to create the product

$$\mathbf{Y} = \mathbf{Q}\mathbf{X}, \quad \text{with } \mathbf{X} \coloneqq [\mathbf{x}_1, \cdots, \mathbf{x}_G] \quad \mathbf{Y} \coloneqq [\mathbf{y}_1, \cdots, \mathbf{y}_G], \tag{16}$$

where in the case of the distributed WF reconstruction the local inputs $\mathbf{x}_i \coloneqq \sigma_i$, the local outputs $\mathbf{y}_i \coloneqq \overline{\mathbf{c}}_i$, and $\mathbf{Q} \coloneqq (\overline{\mathbf{D}}_\mathbf{1}\overline{\mathbf{D}}_\mathbf{1}^\mathrm{T})^{-1}\overline{\mathbf{D}}_\mathbf{1}^\mathrm{T}$ is the system matrix shared by all partitions i. Respectively, for the distributed DM projection, $\mathbf{x}_i \coloneqq \mathbf{c}_i$, $\mathbf{y}_i \coloneqq \mathbf{u}_i$, and $\mathbf{Q} \coloneqq (\mathbf{F}_1(x,y)^\mathrm{T}\mathbf{F}_1(x,y))^{-1}\mathbf{F}_1(x,y)^\mathrm{T}\mathbf{B}_1(x,y)$.

For the local reconstruction, no further adjustments to the D-SABRE algorithm have to be made. The zero padding of the non-illuminated subapertures, introduced to create a square partitioning for the H-PME in Section 3.B, also allows the creation of equally shaped sub-triangulations (see Fig. 8), leading to the identical WFR matrices in Section 2.A. Regular square sub-triangulations arranged in a $2^p \times 2^p$ partition grid are achieved by embedding the partially illuminated square SH array containing N subapertures into an extended square array of $N_\mathrm{ext} \coloneqq (2^p m_p(N) + 1)^2$ subapertures, where factor

$$m_p(N) := \min_{m\in\mathbb{N}}\{m\} \quad \text{such that } 2^p m > \sqrt{N}. \tag{17}$$

The theoretical computational complexity per partition of the D-SABRE method scales then with $\mathcal{O}(N_{\text{ext}}^2/G^2)$flops $= \mathcal{O}(m_p(N)^4)$ flops, because there are a total of $G = (2^p)^2$ partitions. Using the regular Type-II triangulation, as is done throughout this paper, the number of triangles in the core part of each partition is then $J_{\Omega_i} = 4(m_p(N))^2$ (see Fig. 8 and Section 2.A). If an extended array is created according the rule of Eq. (17) to embed the 32×32 SH array used in the CL simulations, one obtains $m_p(N) = 3$ for the very strong 16×16 partitioning and $m_p(N) = 9$ for the moderate 4×4 partitioning.

The distributed DM projection of Section 4.A was slightly changed to not only compute the actuator commands within the pupil but to also here assume an actuator array extending over the entire triangulation. That way, identical DM projection matrices can be enforced for all partitions. Because in this case also the B coefficients located outside of the pupil and computed at least partly from zero slopes are included in the projection, additional errors are potentially created in the computation of the actuator commands at the edges of the pupil. To understand to what extend this adaptation influences the correction quality, the Monte Carlo experiment from Section 4.B (Fig. 12) was repeated, comparing the distributed DM projection based on several MV products respecting the pupil to the matrix-matrix product version just presented. As in the previous CL experiments, it can be seen in Fig. 14 that very strong partitioning has a negative effect on the correction quality. The moderate 4×4 partitioning gives promising results: for moderate noise levels, the D-SABRE with distributed projection adapted for the GPU (MM) achieves Strehl ratios that stay within 0.5% of the correction level obtained with the projection, which is only performed within the pupil (MV). For strong levels, it shows superior noise rejection and outperforms the latter. In comparison with the globally computed SABRE correction, the D-SABRE adapted for the GPU achieves Strehl ratios within 1.2% of the global result for guide star magnitudes $\leq$8 and within 3.6% for magnitudes $\leq$11. This means that a certain trade-off in correction quality has to be made to use the computationally beneficial reformulation of the problem.

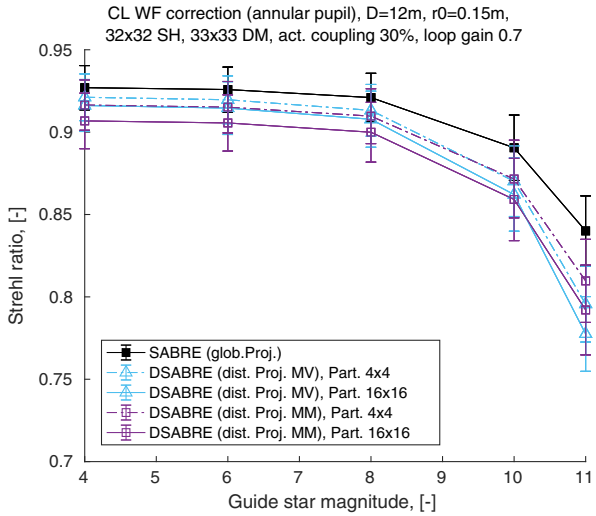

Fig. 14. Correction performance and noise resilience in long-exposure Strehl ratio comparing D-SABRE with distributed DM projection performed only within the pupil (MV) or on the full square pupil plane (MM) for two levels of decomposition and increasing presence of photon shot noise. The WF estimates were computed on the full sensor domain of the zero-padded 32×32 SH array, and the correction was applied within the annular pupil (30% central obscuration). act., actuator; dist. Proj., distributed projection; glob. Proj., global projection; Part., partitioning.

The computational advantages of the presented approach are twofold. First, no dedicated kernel has to be programmed to perform the product, but a single call of a built-in cuBLAS routine that is highly optimized to maximize occupancy and minimize memory latency suffices. Further, the required GPU memory space for storage of the reconstruction and projection matrices, the largest source of data to be allocated on the GPU, is reduced heavily.

The following section describes the GPU implementation of the entire D-SABRE method in further detail. The computational speed is tested by timing runs of the implementation for an example of a very large-scale and an example of an extremely large-scale AO system.

B. Kernel Description and Speed by Timing

The CUDA implementation consists of a row of kernels that are called sequentially and perform the following operations in parallel. First, the local reconstruction is computed with Eq. (6), after which the full local B coefficients are obtained from the projected vector according to Eq. (7). This is followed by the kernels performing the H-PME procedure described in Eqs. (10) and (11), and finally, the actuator commands are computed with the function for the local projection in Eq. (15). Whereas for the local reconstruction and projection, which are performed as MM product adhering to Eq. (16), the parallelization of the computation is executed by the so-called general MM multiplication (GEMM) cuBLAS subroutine, and the kernels of the remaining operations are custom coded to translate prevalent parallelism to the multi-core hardware.

Table 1 lists the CUDA functions that are executed in one run of the D-SABRE method. A name tag identifying the operation is given next to the reference for the respective equation in this paper. It is specified if the function is a cuBLAS routine or a custom-coded CUDA kernel and how many times it is called in one D-SABRE run. In the case of the CUDA kernels, the size of the grid of thread blocks and the number of treads per block are shown in terms of D-SABRE quantities defined throughout the paper.

1. Custom-Coded Kernels

Because the D-SABRE considers splines of polynomial degree $d = 1$ and continuity order $r = 0$, the expansion of the null space of the local constraint matrices in Eq. (7) reduces to a resorting of the projected B-coefficient vector of size $\overline{d}_i$ into a larger vector of $J_i\hat{d}$ expanded B coefficients within all G partitions.

- This index swap is a partition local operation, which allows independent execution in G thread blocks, with the index

Table 1. Kernel Overview

Operation	Eqs.	Type	Grid	Block	Calls	Execution Time (Total/Iteration) ($\sqrt{N}=100, p=4$)	Execution Time (Total/Iteration) ($\sqrt{N}=200, p=5$)
Loc. reconstruction	(6), (16)	cuBLAS	–	–	1	86 µs	168 µs
Exp. null space	(7)	Kernel	$G=2^{2p}$	$J_i\hat{d}$	1	22 µs	67 µs
Comp. differences	(10)	Kernel	$\gamma 2^{2p-h-1}$	$2m_p(N)$	$2p$	48 µs	64 µs
Comp. offsets	(10)	Kernel	1	$\gamma 2^{2(p-h)}$	$2p$	37 µs	46 µs
Offset partitions	(11)	Kernel	$\gamma 2^{2(p-1)}$	$J_i\hat{d}$	$2p$	55 µs	166 µs
Loc. projection	(15), (16)	cuBLAS	–	–	1	130 µs	147 µs
D-SABRE (+ proj.)	–	–	–	–	1	379 µs	658 µs
Memory copy	(6)	HostDevice	–	–	1	200 µs	862 µs
Memory copy	(15)	DeviceHost	–	–	1	10 µs	33 µs

computation and assigning of the to-be-expanded coefficients performed cooperatively but in parallel in the threads. Recalling the earlier mentioned limitation of numbers of threads per block, the decomposition of the triangulation has to be strong enough to guarantee that $J_i\hat{d} < 1024$.

The H-PME procedure constitutes three kernels of different grid and block sizes. As introduced in Section 3.A, each level $h = 1, \cdots, p$ performed for a $2^p \times 2^p$ partitioning, eliminates the piston offsets between four tiles of partitions that are grouped together. The number of groups and the number of partitions per tile depend hereby on level h and power p (see Section 3.A). Because the computation of H-PME offsets in Eq. (10) involves coefficients shared between the partition tiles, parallelism had to be exploited along these edges of the tiles and is realized in two kernels.

- The computations of the averaged differences of coefficients located on the considered partition edges, i.e., the addends of the sum in Eq. (10), are partition local operations. The first H-PME kernel assigns each addend computation to an independent thread block. The subtractions of the edge coefficients are performed on the threads, evoking additional fine-grained parallelism.
- For the computation of the actual offsets in the second kernel, no coarse-grained data parallelism could be achieved, because the sum in Eq. (10) combines data allocated to various partitions. The kernel is therefore limited to parallelizing the operation in a cooperative manner, with each offset computation being assigned to a thread within the single thread block.
- The actual offsetting of the local B coefficients is parallelized in the third kernel in the straightforward manner of linking the offset partitions to blocks and the coefficients to threads.

Because the equalization of the piston modes within the groups of four tiles requires a synchronization between the thread blocks after the first three tiles are leveled, each H-PME level is performed by two calls of the three presented kernels. This affects also the grid and block sizes, which are given for the case of a square pupil in Table 1. For the first call of the H-PME kernels in each level h, the constant $\gamma = 2$; for the second call, $\gamma = 1$.

2. Kernel Execution and Memory Transfer Times

The two last columns in Table 1 give speed by timing of the presented CUDA implementation of the D-SABRE performed on the hardware specified in Table 2. The implementation was tested for a very large-scale SH array of $N = 10^4$ subapertures and an extremely large-scale array with $N = 4 \cdot 10^4$, which were embedded in grids of $2^p \times 2^p$ sub-triangulations according to Eq. (17), with $p = 4$ and $p = 5$, respectively. With these moderate partitionings, one obtains $m_p(N) = 7$ for both scenarios, which results in the same theoretical computational complexity of $\mathcal{O}(m_p(N)^4)$ flops per partition. However, the number of partitions increases from $G = 256$ for the very large-scale case to $G = 1024$ for the extremely large-scale case. An annular pupil with a central obscuration covering 30% of the area was assumed and a Fried geometry chosen for the layout of the actuator grid. The speeds by timing, given in microseconds, are averages of kernel execution times obtained with the CUDA profiling tool NVPROF [26] from 10 runs of the CUDA code performing the D-SABRE method, and they indicate the total times consumed by the kernel calls in one D-SABRE run.

For the $p = 4$ case, the local reconstruction and the local projection bear the longest computation times; for the larger partition grid of $p = 5$, the $2p$ calls of the kernel that performs the H-PME partition offsets constitute a similarly time-intensive part of the implementation. It also shows that the cooperative calculation of the H-PME offsets, which is performed in a single block because information from partitions has to be shared, did not create a bottleneck in the implementation and remains one of the kernels with the lowest total execution time for both cases. The overall kernel execution time per D-SABRE run (including DM projection) stands at 379 µs for the very large-scale and at 658 µs for the extremely large-scale AO system. To give an idea of the speedup achieved by implementing the D-SABRE on the GPU, the so-called GEMM BLAS subroutine of the Atlas library was used to perform *solely* the MM product of the local reconstruction. The processing time of the function was measured with the system-wide real-time clock. With the CPU listed in Table 2, the local WFR alone required 5 ms for the very large-scale and 18.5 ms for the extremely large-scale scenario.

Because memory latencies can create a major bottleneck in GPU computing, Table 1 also lists the times that are spent in each iteration on data transfers between the host (i.e., CPU) and the device (i.e., GPU) memory via the PCI Express interface [25]. To reduce such memory copies to a minimum, the D-SABRE implementation contains several C routines that generate all precomputable data that are necessary for the

Table 2. Hardware Overview

CPU:	Intel(R) Xeon(R) CPU E5-1620 0
Cores	4
Threads	8
Processor base frequency	3.60 GHz
System memory (RAM)	16 GB
GPU:	**GeForce GTX TITAN X**
Micro architecture	Maxwell
Base clock	1.09 GHz
CUDA cores	24 × 128
Memory bandwidth	336 GB/s
GPU memory	12 GB DDR5
System interface	PCI Express 3.0 × 16
Compute capability	5.2

real-time CUDA kernels. Using Unified Memory [25] to simplify the code, these data are allocated, declared, and defined directly on the device memory, and only structures of pointers to the required data locations are transferred to the real-time kernels of Table 1. In each iteration, the D-SABRE measurement vectors σ_i from Eq. (6) are, stored in a global vector, transferred from the host to the device in a single memory copy. The reverse copy is performed at the end of each iteration for a stacked vector of the actuator commands $\mathbf{u}_i$ obtained from Eq. (15). Despite these efforts, the time spent on memory transfer via the PCI Express outweighs the overall kernel execution time for the $p = 5$ case. The use of more powerful interconnect systems like NVIDIA NVLink [27], which are currently introduced to the market, would provide immediate speedup of memory transfer.

Including the needed host–device communication, the presented CUDA implementation of the D-SABRE method enables the user to compute the DM actuator command updates in 0.59 ms for a very large-scale AO system with a SH array of $N = 10^4$ subapertures and in 1.55 ms for an extremely large-scale system of $N = 4 \cdot 10^4$. With a standard off-the-shelf GPU, computation times that are edging toward the kilohertz update frequencies, targeted for the benchmark AO system of the E-ELT, were achieved.

6. CONCLUSIONS

We present a fully distributed algorithm, based on the D-SABRE method, for WF correction in extremely large-scale AO systems. The method is intended for SH slope measurements and the execution on parallel hardware.

The D-SABRE method for WFR is constructed with a B-spline model of the WF. The local nature of the B-spline basis functions allows the decomposition of the WFS domain into partitions on which the WFR is locally performed in a distributed manner. The procedure for equalization of the unknown local piston modes of the original version of the method, the D-PME, showed incompatibility with large central obscurations and suffered from error propagation for large numbers G of partitions. The presented hierarchical H-PME fixes these issues by creating information exchange not only between directly neighboring partitions but between groups of partitions through a multi-level approach. This allows application to extremely large-scale AO systems, where the number of partitions G has to be set sufficiently large to adequately distribute the computational load. The hierarchical leveling of the partitions with the H-PME also allows faster convergence in only $\log_2(\sqrt{G})$ iterations compared to $\sqrt{G}/2$ iterations required with the sequential information flow of the D-PME, and it shows superior noise-rejection properties in numerical experiments with the OOMAO simulation tool.

To compute the DM actuator commands from the SH data in a fully distributed manner, the projection of the B coefficients, which describe the WF estimates, onto the space of actuator commands was formulated locally for each partition. The intercoupling of actuators located at the partition edges is, to a certain extent, taken into account by partition overlap. In simulation, the procedure has provided stable long-exposure Strehl ratios for actuator couplings of 30% or lower at varying loop gains.

An implementation of the described distributed WF correction method, based on the D-SABRE with H-PME, for the GPU was programmed with the parallel computing platform NVIDIA CUDA. The algorithm was adapted to the hardware by enforcing identical sub-triangulations, which allows reformulating the computationally most expensive operations, i.e., the local WF reconstructions and the local DM projections, to a MM product. This so-called compute-bound operation is prone to significant speedup if executed on a GPU and can be performed with the highly optimized GEMM cuBLAS subroutine. Several custom-coded CUDA kernels that execute the H-PME and translate prevalent parallelism to the multicore structure of the GPU complete the implementation. Speed tests by timing for single runs of the method were realized with a standard GPU. They include, next to the execution time of all CUDA kernels, the low-bandwidth host–device data transfers, which could be reduced to a single copy of the SH data vector and the command vector per run: the CUDA implementation of D-SABRE correction method accomplishes the actuator command update with 0.59 ms for a very large-scale AO system of $N = 10^4$ and 1.55 ms for an extremely large-scale test of $N = 4 \cdot 10^4$, indicating linear scaling of the D-SABRE update time with N.

To obtain the computationally beneficial version of the D-SABRE method presented in this paper, a certain trade-off in reconstruction accuracy has to be made. The H-PME procedure requires a square grid of $2^p \times 2^p$, $p \in \mathbb{N}$, sub-triangulations, which have to be of identical size and shape to allow the realization of the local WF reconstructions and DM projections as MM products. To create applicability to arbitrary pupil shapes and SH array dimensions, the illuminated subapertures are embedded in a square SH array of suitable dimension, and zero slopes are processed for the non-illuminated subapertures. Local reconstruction errors that occur due to this zero padding in partitions located at the edges of the pupil have to be addressed in future work, also in view of the inclusion of telescope spiders. Further efforts should be undertaken to extend the distributed DM projection to very strong actuator coupling through exact solution. For low to medium coupling, the current local approximation provides superior Strehl ratios if compared with the global

DM projection, because local WFR errors are not propagated throughout the grid of actuator commands.

Whereas the D-SABRE method was devised for SH sensor measurements, we are aware of the shift toward the pyramid WFS (P-WFS) [28,29] as baseline for, among others, the eXtreme AO system on the planned E-ELT [2], and future work will be dedicated to this matter. An immediate extension of the D-SABRE to P-WFS measurements can be achieved with a preprocessing step presented by Shatokhina *et al.* [30]. The suggested transformation of P-WFS data to SH data is of $\mathcal{O}(N)$ computational complexity and highly parallelizable and would therefore not affect the scalability of the D-SABRE method.

Funding. Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) (2009/09093/BOO); European Strategy Forum on Research Infrastructures (ESFRI).

Acknowledgment. The authors thank Niels Tielen, whose M.Sc. project laid the foundation for this collaborative work. Kees Lemmens played a crucial role in the implementation of the algorithm for the GPU and receives the authors' special thanks for continuous support throughout the project.

REFERENCES

1. F. Roddier, *Adaptive Optics in Astronomy* (Cambridge University, 1999).
2. V. Korkiakoski and C. Vérinaud, "Simulations of the extreme adaptive optics system for EPICS," Proc. SPIE **7736**, 773643 (2010).
3. 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).
4. B. L. Ellerbroek, "Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques," J. Opt. Soc. Am. A **19**, 1803–1816 (2002).
5. L. Gilles, C. R. Vogel, and B. L. Ellerbroek, "Multigrid preconditioned conjugate-gradient method for large-scale wave-front reconstruction," J. Opt. Soc. Am. A **19**, 1817–1822 (2002).
6. D. G. MacMartin, "Local, hierarchic, and iterative reconstructors for adaptive optics," J. Opt. Soc. Am. A **20**, 1084–1093 (2003).
7. P. J. Hampton, P. Agathoklis, and C. Bradley, "A new wave-front reconstruction method for adaptive optics systems using wavelets," IEEE J. Sel. Top. Signal Process. **2**, 781–792 (2008).
8. E. Thiébaut and M. Tallon, "Fast minimum variance wavefront reconstruction for extremely large telescopes," J. Opt. Soc. Am. A **27**, 1046–1059 (2010).
9. M. Rosensteiner, "Cumulative reconstructor: Fast wavefront reconstruction algorithm for extremely large telescopes," J. Opt. Soc. Am. A **28**, 2132–2138 (2011).
10. C. C. de Visser, E. Brunner, and M. Verhaegen, "On distributed wave-front reconstruction for large-scale adaptive optics systems," J. Opt. Soc. Am. A **33**, 817–831 (2016).
11. C. C. de Visser and M. Verhaegen, "Wavefront reconstruction in adaptive optics systems using nonlinear multivariate splines," J. Opt. Soc. Am. A **30**, 82–95 (2013).
12. M. J. Lai and L. L. Schumaker, *Spline Functions on Triangulations* (Cambridge University, 2007).
13. D. L. Fried, "Least-square fitting a wave-front distortion estimate to an array of phase-difference measurements," J. Opt. Soc. Am. A **67**, 370–375 (1977).
14. M. Rosensteiner, "Wavefront reconstruction for extremely large telescopes via CuRe with domain decomposition," J. Opt. Soc. Am. A **29**, 2328–2336 (2012).
15. R. V. Shack and B. C. Platt, "Production and use of a lenticular Hartmann screen," J. Opt. Soc. Am. **61**, 656–660 (1971).
16. M. Viegers, E. Brunner, O. Soloviev, C. C. de Visser, and M. Verhaegen, "Nonlinear spline wavefront reconstruction through moment-based Shack–Hartmann sensor measurements," Opt. Express **25**, 11514–11529 (2017).
17. E. Brunner, C. C. de Visser, and M. Verhaegen, "Nonlinear spline wavefront reconstruction from Shack–Hartmann intensity measurements through small aberration approximations," J. Opt. Soc. Am. A **34**, 1535–1549 (2017).
18. C. C. de Visser, Q. P. Chu, and J. A. Mulder, "Differential constraints for bounded recursive identification with multivariate splines," Automatica **47**, 2059–2066 (2011).
19. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn. **3**, 1–122 (2010).
20. R. Conan and C. Correia, "Object-oriented Matlab adaptive optics toolbox," Proc. SPIE **9148**, 91486C (2014).
21. E. Vernet, M. Cayrel, N. Hubin, R. Biasi, D. Gallieni, and M. Tintori, "On the way to build the M4 unit for the E-ELT," Proc. SPIE **9148**, 914824 (2014).
22. R. Ellenbroek, M. Verhaegen, R. Hamelinck, N. Doelman, M. Steinbuch, and N. Rosielle, "Distributed control in adaptive optics—deformable mirror and turbulence modeling," Proc. SPIE **6272**, 62723K (2006).
23. R. Gupta, D. Lukarski, M. B. van Gijzen, and C. Vuik, "Evaluation of the deflated preconditioned CG method to solve bubbly and porous media flow problems on GPU and CPU," Int. J. Numer. Methods Fluids **80**, 666–683 (2016).
24. H. Knibbe, C. Vuik, and C. W. Oosterlee, "Reduction of computing time for least-squares migration based on the Helmholtz equation by graphics processing units," Comput. Geosci. **20**, 297–315 (2016).
25. NVIDIA Corporation, "NVIDIA CUDA C programming guide," Version 4.2 (2012).
26. NVIDIA Corporation, "Profiler user's guide," Version 5.5 (2013).
27. D. Foley and J. Danskin, "Ultra-performance Pascal GPU and NVLink interconnect," IEEE Micro **37**, 7–17 (2017).
28. R. Ragazzoni, "Pupil plane wavefront sensing with an oscillating prism," J. Mod. Opt. **43**, 289–293 (1996).
29. R. Ragazzoni, E. Diolaiti, and E. Vernet, "A pyramid wavefront sensor with no dynamic modulation," Opt. Commun. **208**, 51–60 (2002).
30. I. Shatokhina, A. Obereder, M. Rosensteiner, and R. Ramlau, "Preprocessed cumulative reconstructor with domain decomposition: A fast wavefront reconstruction method for pyramid wavefront sensor," Appl. Opt. **52**, 2640–2652 (2013).