Exploiting Symmetries and Progressive Refinement for Apodized Pupil Lyot Coronagraph Design

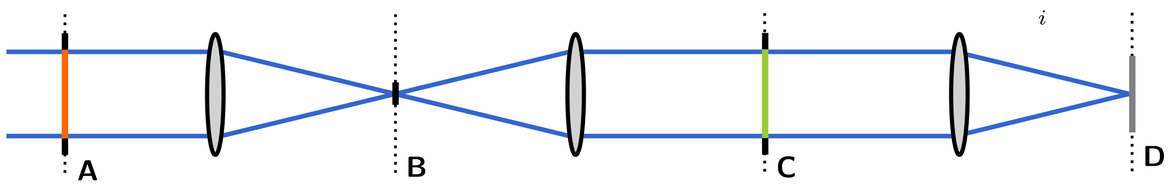
Emiel H. Por[1], Rémi Soummer[2], James Noss[2] & Kathryn St. Laurent[2]

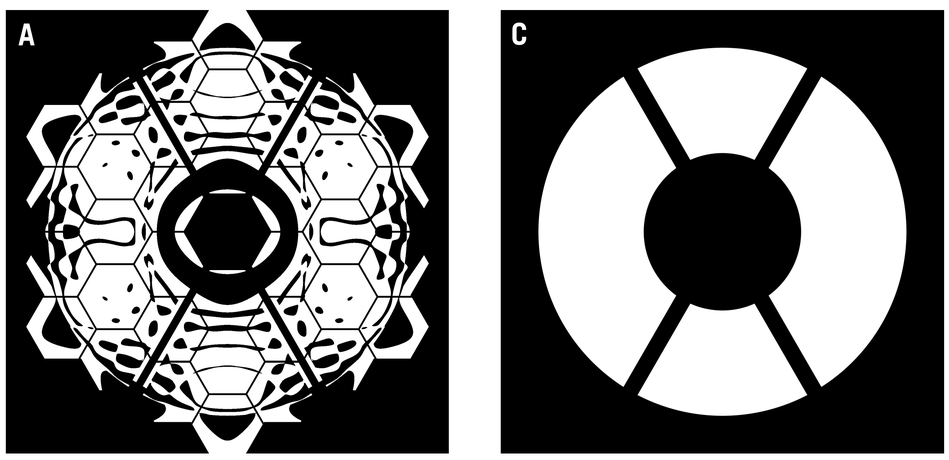
[1] Leiden Observatory, The Netherlands; [2] Space Telescope Science Institute, USA

Contact: por@strw.leidenuniv.nl

Goal

Modern coronagraph design relies on advanced, large-scale optimization processes that require an ever increasing amount of computational resources. We explore ways to reduce this large computational demand and allow for higher-resolution optimizations to be performed for the Apodized Pupil Lyot coronagraph. We exploit symmetries and use progressive refinement and reduce the memory consumption by >256 times.



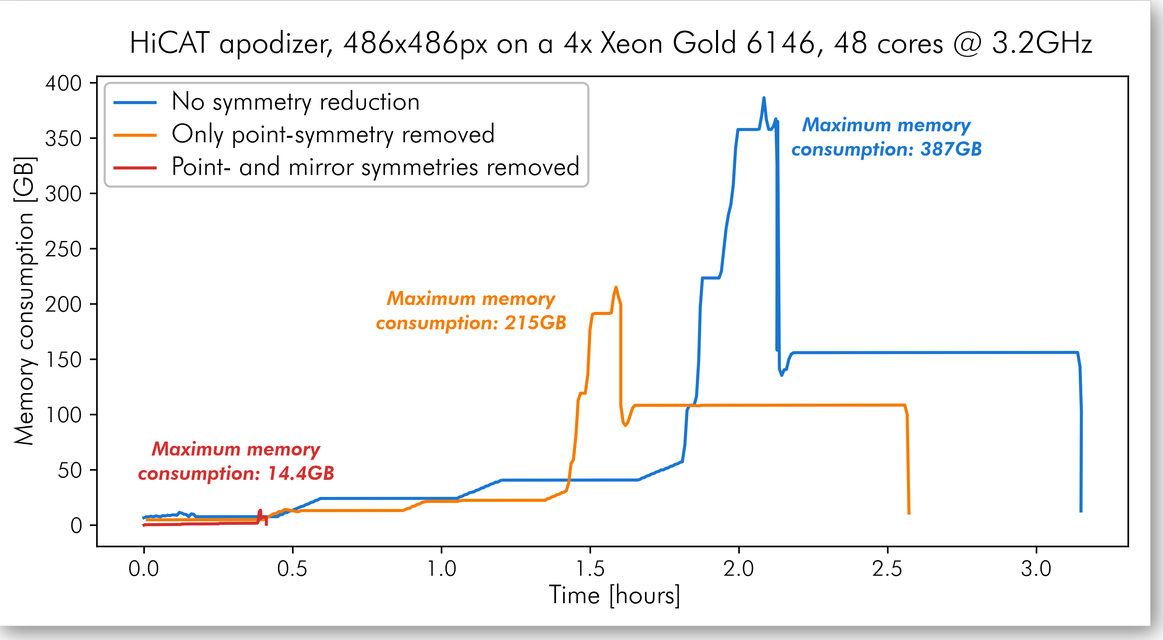


Point-reflection symmetry

This symmetry occurs when we have a point-symmetric focal-plane mask and dark zone. In this case, when we have found a solution for the apodizer, its conjugate is also a solution. Therefore we can remove half of the variables (all imaginary parts of the apodizer) and half of the constraints (half of the dark zone).

Mirror symmetries

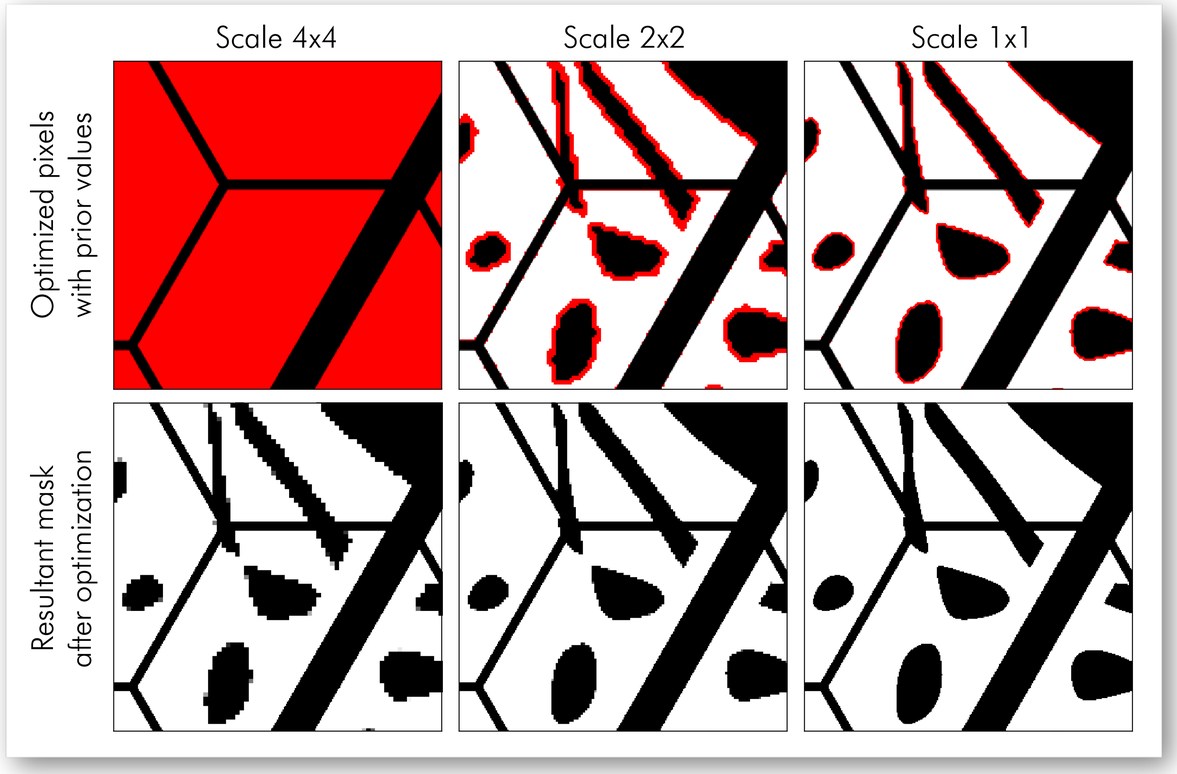
This symmetry occurs when we have mirror symmetry in the pupil and Lyot stops. Each mirror symmetry removes half of the variables (halving the apodizer along the mirror symmetry line) and half of the constraints (either quartering the dark zone or removing imaginary parts if the dark zone is already quartered).

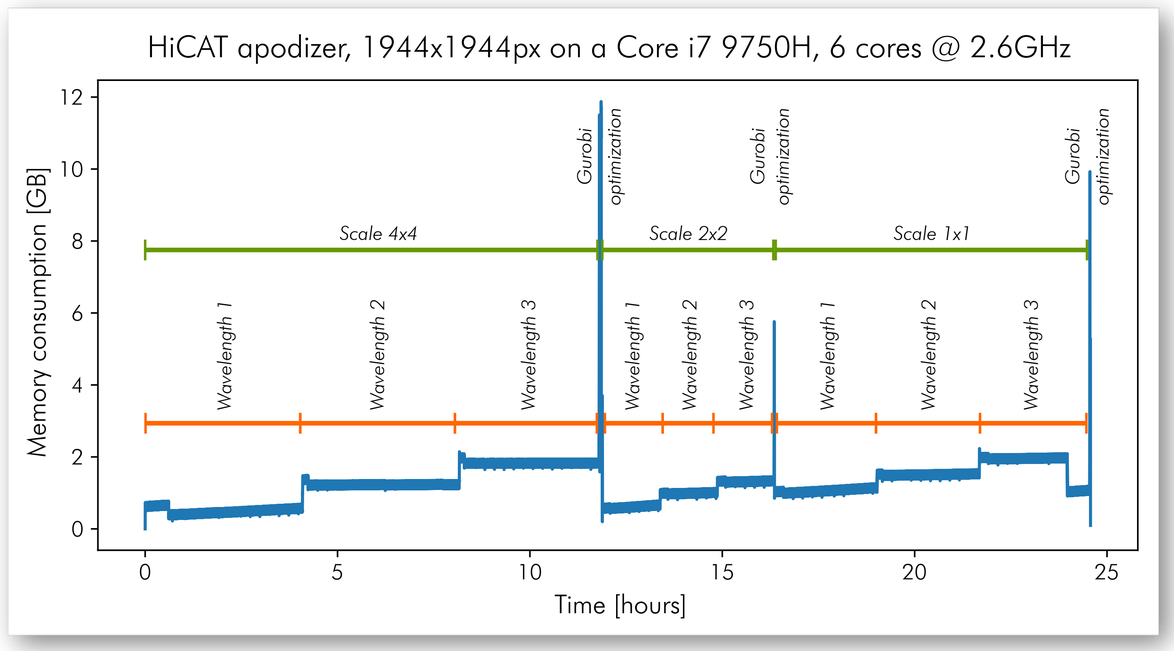


Point symmetry removed?	Mirror symmetry removed?	# variables	# constraints	Reduction in memory consumption
✗	✗	141,716	31,872	0%
✓	✗	141,716	15,936	~50%
✓	✓	35,429	3,984	~97%

Progressive refinement

Lower resolution apodization solutions can act as a prior on the opaque and transmissive regions in a higher-resolution solution. This allows us to not optimize any pixels that are "clearly" either opaque or transmissive and use our variables for pixels that are still "unclear". Additionally, the number of variables now scales as the square root of the number of pixels in the mask, rather than linearly.





Scale	# variables (progressive)	# variables (non-progressive)	# constraints	Reduction in memory consumption
4x4	35,429	35,429	3,984	0%
2x2	14,213	141,716	3,984	~90%
1x1	25,691	566,864	3,984	~95%

Conclusions

We have shown two ways of reducing memory consumption and runtime when designing the apodizer for an apodized-pupil Lyot coronagraph. Removing symmetries that this optimization problem exhibits reduces memory consumption by **32×**. Using progressive optimization yields an additional **~22×** reduction. Together, we achieved a reduction of **~706×** in memory consumption. These improvements allow us to optimize a 1944x1944px apodizer on a laptop with 16GB of memory, which would have required a cluster with >7TB of memory otherwise.