Performance Comparison of Optimization Methods for Blind Deconvolution

Daniel Thompson
The Boeing Company
550 Lipoa Parkway
Kihei, HI 96753
daniel.o.thompson2@boeing.com

Brandoch Calef
The Boeing Company
550 Lipoa Parkway
Kihei, HI 96753
brandoch.calef@boeing.com

Michael Werth
The Boeing Company
550 Lipoa Parkway
Kihei, HI 96753
michael.p.werth@boeing.com

Abstract— **There are many methods that will solve high-dimensional regression problems, and choosing an appropriate method can be challenging. For some problems, accuracy holds precedence over speed whereas in other instances speed is required for a large number of problem sets. In this paper we study the performance of several methods that solve the multiframe blind deconvolution problem by comparing speed and accuracy of each algorithm, highlighting the merits of each algorithm.**

Table of Contents

1. Introduction

Multi-frame blind deconvolution (MFBD) [1] is a problem in which the true image and blurring function are estimated from their convolution using multiple variably blurred images of the same object. One application of MFBD is the recovery of images of space objects observed from the ground through atmospheric turbulence. Over several short-exposure frames, a stable satellite will change minimally and the frame-to-frame differences can be attributed to changing point-spread functions (PSFs). A more in depth look at the history and advantages of MFBD have been described previously [2]. The main objective of this paper is to examine the suitability of different algorithms in solving the MFBD maximum likelihood optimization problem

$$\arg\min_{o,p\geq 0} \sum_{i,j} \frac{(o * p_i - d_i)_j^2}{(o * p_i)_j + \sigma^2}, \tag{1}$$

where o is the object estimate, p_i is the ith PSF estimate, d_i is the ith frame of noisy measured data, and σ^2 is the read noise variance in each pixel. The notation $(d_i)_j$ (for example) denotes the jth pixel of the ith data frame.

Numerous methods exist that could be applied to problems such as (1), and it is not often clear which is the best to use for a given problem. This paper examines several of these algorithms, comparing the speed and results of the reconstructions of simulated space object imagery.

978-1-4673-7676-1/16/$31.00 ©2016 IEEE

2. Packages Examined

The algorithms tested below are drawn from a number of publicly-available software packages: L-BFGS-B [3], NLopt [4], NLCG [5], Poblano [6], and ASP [7]. Some of the packages offer multiple algorithms, and in some cases there is overlap from one package to another, but there are differences in implementation that make for interesting comparisons. All the algorithms under consideration are iterative nonlinear, gradient-based local optimization methods. The only real outlier is ASP, which is discussed in more detail below.

For all the algorithms examined, except the Active-Set Basis Pursuit (ASP) [7], [8] package, the positivity constraint on the estimated image and PSF pixels is enforced within the gradient calculation, where the gradient function is provided by the MFBD algorithm. This constraint is enforced by one of two schemes. The first is to reformulate the estimation problem so that the estimated quantities are $\sqrt{o}$ and $\sqrt{p_i}$. The second is a penalized scheme where we adjust the gradient to push negative values in the positive direction. For the ASP package, providing a gradient function was not an option and we instead relied on the algorithm enforcing this positivity constraint itself. Some algorithms also allowed the imposition of bound constraints and those options were tested.

Packages and Methods

This is a brief list of the packages and methods along with acronyms:

1. *LBFGSB*: Limited-memory BFGS with bound constraints package
2. *NLopt*: Nonlinear optimization package [4]
   (a) Limited-memory BFGS (LBFGS)
   (b) Truncated Newton method (TNM)
   (c) Conservative convex separable approximation (CCSA)
   (d) Method of moving asymptotes (MMA)
   (e) Shifted limited-memory variable-metric (SLMVM)
3. *Poblano* MATLAB toolbox [6]
   (a) Limited-memory BFGS (LBFGS)
   (b) Truncated Newton method (TNM)
   (c) Nonlinear conjugate gradient (NCG)
4. *NLCG*: Nonlinear conjugate gradient package [5]
5. *ASP*: Active set package [7], [8]

LBFGSB

LBFGSB [3] implements limited-memory BFGS with optional bound constraints. This code is FORTRAN converted to C and compiled into a MATLAB mex file.

NLopt

NLopt [4] is a compilation of different algorithms, written in multiple languages, by different authors, and uses a MATLAB wrapper as the upper level interface.

Poblano

Poblano [6] is written entirely in MATLAB and this version contains optional inputs, such as the search direction method, conjugate gradient stopping tolerance, and other tolerance values. Only minimal differences were observed when changing these options.

NLCG

NLCG [5] is a collection of nonlinear conjugate gradient algorithms. This package is implemented in MATLAB.

ASP

The ASP [7] package contains several methods that will solve least squares problems, none of which can address nonlinear problems such as (1). Instead, the problem is linearized by alternately fixing o and $\{p_i\}$. In addition, an L_1 regularization term is added, yielding the new minimization problem

$$\arg\min_o \left(\sum_{i,j} \frac{(o * p_i - d_i)_j^2}{d_{ij} + \sigma^2} + \lambda \sum_j |o_j| \right) \tag{2}$$

and likewise for $\{p_i\}$. The ASP package is written entirely in MATLAB.

3. Algorithms Examined

Truncated Newton

The truncated Newton method [9], [10], [11] uses a conjugate gradient algorithm to approximately solve the Newton equations to obtain a new search direction. Then it performs a line search to the next solution. This method is less expensive computationally than solving the Newton equations exactly and has been shown to yield accurate results in large problems. Generally, this method does not require the Hessian matrix explicitly, which reduces storage requirements. The methods examined here use finite differencing to approximate the Hessian.

Each package considered contains various options to tweak the algorithm's behavior. In our own experiments the options appeared to have minimal impact on the reconstruction, and the results from the default options are presented in this paper. In addition, each package performs a line search based on the Moré-Thuente method [12]. This is to guarantee global convergence, meaning convergence to some local minimum is guaranteed, rather than the global minimum.

The NLopt truncated Newton algorithm is written in C, which has been translated from FORTRAN using f2c with minor changes. It contains several variations of the truncated Newton method, allowing for search direction restarting and preconditioning. The variant examined here is the basic algorithm with no restarting and no preconditioning. In preliminary test cases, only minimal differences were observed when changing these options.

Limited-memory BFGS

Limited-memory Broyden-Fletcher-Goldfarb-Shanno [13] is an unconstrained quasi-Newton method. As the name suggests, it is a variant of the BFGS algorithm that employs a low-rank approximation of the Hessian (the full Hessian for the blind deconvolution problem would be unmanageably large).

The NLopt version was written in FORTRAN and converted to C using f2c with minor modifications. It heuristically determines the number of previous LBFGS corrections to be kept in history. NLopt will also attempt to heuristically choose the step length based on bounds, tolerances and other information.

The Poblano version uses a two-loop recursion [14] to approximate Hessian-gradient products and the Moré-Thuente line search. One of the LBFGS-specific options is the number of previous iterations to keep in history. Changing this value over a reasonable range appeared to have minimal effects on the reconstruction.

The L-BFGS-B [15], [16] version of LBFGS is a constrained quasi-Newton method and can enforce bound constraints when minimizing (1). When allowing it to enforce positivity itself vs. using the MFBD scheme, no improvement was observed. No upper bound was set in the examined data of this paper.

Conservative Convex Separable Approximation

NLopt is the only package examined that implements this method [17] and identifies its particular variant as CCSAQ due to its approximations focusing on simple quadratic terms. This algorithm is written in C with a MATLAB wrapper. A benefit to this method is that it can be applied to large problems in which the Hessians are dense. The algorithm calculates the next iterate using an approximation to the original objective and constraint functions. CCSAQ performs this approximation with simple quadratic terms. If certain bound conditions are satisfied at the candidate iterate, then it is taken as the next iterate solution. Otherwise it is rejected and the next iterate is recalculated with an increased penalty term on the quadratic approximation.

Method of Moving Asymptotes

NLopt is the only package examined that implements this method [17]. This algorithm is written in C with a MATLAB wrapper and is based on the CCSA method. This is a more modern version of MMA that is globally convergent and faster than the original MMA algorithm. The algorithm calculates the next iterate using an approximation to the original objective and constraint functions. MMA performs this approximation with the gradient of the objective function and a quadratic penalty term. If certain bound conditions are satisfied at the candidate iterate, then it is taken as the next iterate solution. Otherwise it is rejected and the next iterate is recalculated with an increased penalty term.

Shifted Limited-Memory Variable-Metric

NLopt is the only package examined that implements this method. It is based on a FORTRAN implementation converted to C with f2c. Variable metric methods [11], [18] are a class of quasi-Newton methods. In this implementation the Hessian can be updated using a rank-1 formula or a rank-2 formula, and both are examined in this paper. Both maintain symmetry in the Hessian but the rank-2 will also maintain positive definiteness. This algorithm is similar to the LBFGS method, which implements a rank-2 update. The "shifted" portion refers to how the Hessian is defined, which is different from LBFGS, and involves an additional term.

Nonlinear Conjugate Gradient

Nonlinear conjugate gradient (NCG) algorithms tend to follow the general format of:

- Evaluate gradient $\nabla f(x_n)$ of objective function at the current position x_n
- Obtain a new search direction $s_n = -\nabla f(x_n) + \beta s_{n-1}$, where β depends on the flavor of NCG
- Perform a line search: $\alpha_n = \arg\min_\alpha f(x_n + \beta s_n)$
- Update position: $x_{n+1} = x_n + \alpha_n s_n$

There are several variations of NCG depending on the choice of β and the line search algorithm.

NLCG provides numerous flavors and we examine the following: Fletcher-Reeves [19], Polak-Ribiere-Polyak [20], steepest descent [14], Polak-Ribiere-Polyak constrained by Fletcher-Reeves, Hager-Zhang [21], and Dai-Yuan [22]. It can also perform the Hestenes-Stiefel method [23], but this variation quits very early and is not included in this study. All these variants utilize a line search method that requires the strong Wolfe condition [24], [25] for convergence purposes, except the Dai-Yuan variant, which allows for a weak Wolfe line search as well as strong. However, the weak Wolfe variant is not examined in this paper.

Poblano provides the Fletcher-Reeves, Polak-Ribiere-Polyak, Hestenes-Stiefel, and steepest descent variations. This package contains an optional conjugate direction restart strategy based on the number of iterations or the orthogonality of gradients across iterations. Adjusting this feature has had minimal effects on the final reconstruction and the results presented in this paper do not use this restart option.

ASP

ASP solves regularized nonnegative least-squares problems by solving the dual basis pursuit problem [8]. An active set method designates each element in the current iterate as "active" or "inactive" based on a set of constraints. Generally elements are set to "inactive" when the algorithm determines an element is an optimal value, and works on the "active" elements. This approach reduces the dimensionality of the problem. This package has its own positivity scheme and does not use the two examined with the other packages. It also requires the PSF and image to be estimated separately, so we experiment with the total function evaluations used to generate the image as well as the total number of evaluations between calculating the image and PSF. In addition there are several tolerance and other algorithm options. Most had minimal effects on the data when adjusted. One option is included in the data. The "homotopy" option is examined because in edge cases the default options would fail to enforce positivity, but with the "homotopy" option the positivity constraint remained enforced.

4. Results

The input data used in these experiments is a CAD model of a satellite convolved with simulated atmospheric PSFs. The stopping criterion for all experiments is the number of function evaluations. Each algorithm was run for 100, 200, 300, 400, and 500 function evaluations, or as close as possible as some algorithms can require several function evaluations to complete an iteration. We recorded the final value of the objective function (1) and the time it took for each set of evaluations. These data are plotted against the

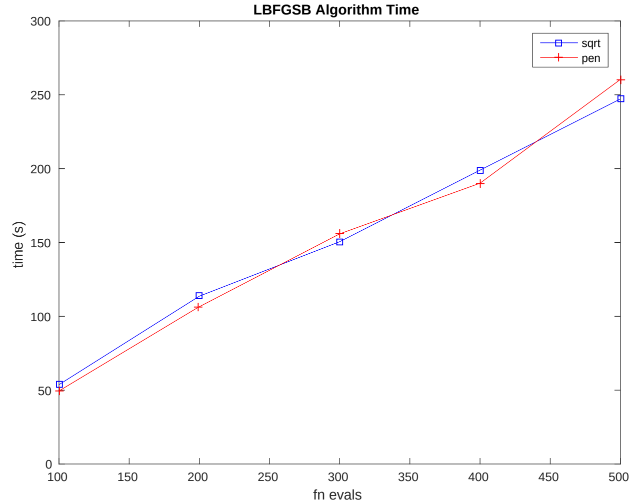

Figure 1. The two positivity schemes require approximately the same amount of time.

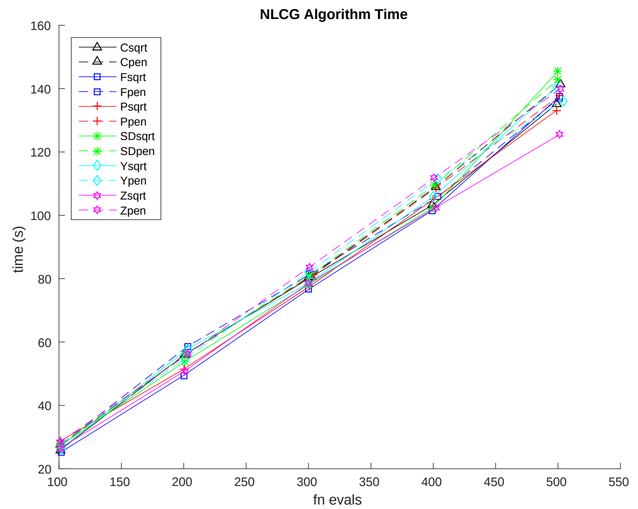

Figure 2. F is Fletcher-Reeves. P is Polak-Ribiere-Polyak. C is P constrained by F. SD is steepest descent. Y is Dai-Yuan. Z is Hager-Zhang.

number of function evaluations and compared to each other. In the legends of the plots, "sqrt" refers to the square root positivity scheme and "pen" refers to the penalized positivity scheme. Also shown here are selected reconstructions from each algorithm/package.

Time Results

Figures 1–5 are plots of the number of function evaluations vs. the time in seconds to complete. The plots are organized by package.

In terms of execution time per function evaluation, we see that the NLCG, NLopt, and Poblano packages tend to be the fastest. Note that this is really just a measure of how much time each algorithm requires internally; the convergence rate has not been taken into consideration in these plots. Comparing Poblano and NLopt, we see that although NLopt is using compiled C code, it does not have a speed increase over Poblano which is written entirely in MATLAB. We also see that between the two positivity schemes there

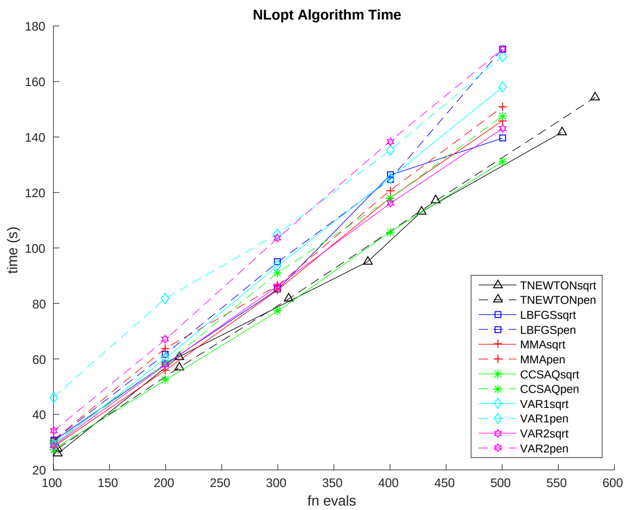

Figure 3. The penalized positivity scheme generally takes more time than the square root positivity scheme

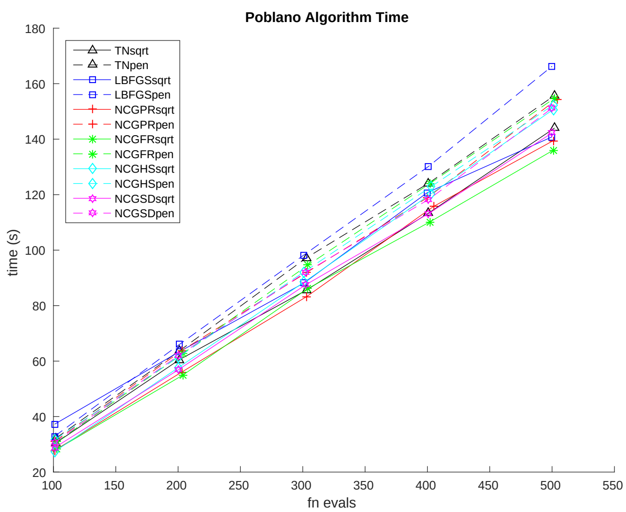

Figure 4. TN is Truncated Newton. NCGPR is Polak-Ribiere-Polyak. NCGFR is Fletcher-Reeves. NCGHS is Hestenes-Stiefel. NCGSD is steepest descent.

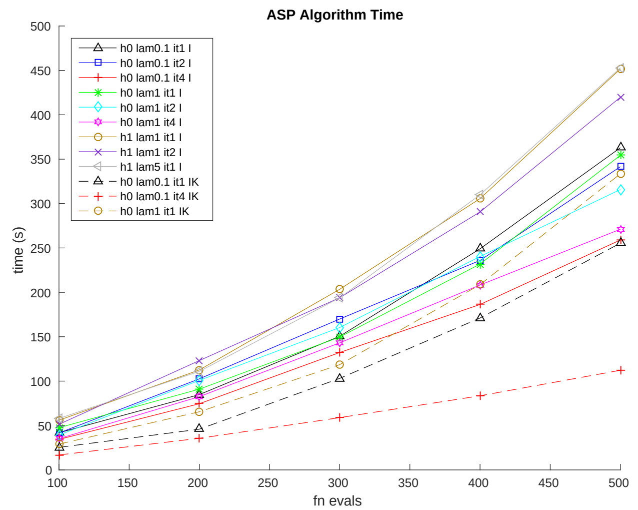

Figure 5. "h0" refers to the homotopy option turned off. "h1" refers to the homotopy option turned on. "lam" refers to the regularization parameter λ used. "it" refers to the number of iterations the function evaluations were split over. "I" refers to only image function evaluations counted. "IK" refers to both image and kernel function evaluations counted.

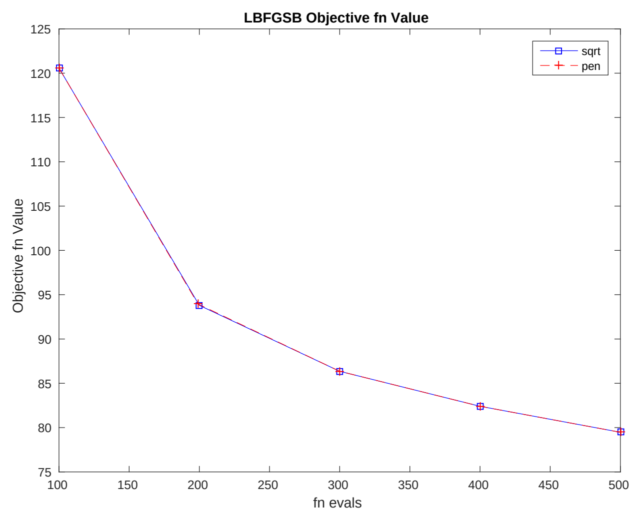

Figure 6. The two penalization schemes yield approximately the same objective function value.

is no consistent difference in speed, and the times taken to process are generally the same.

Objective Function Results

Figures 6–10 are plots of the number of function evaluations vs the objective function value after that many evaluations. The plots are organized by package.

From the Figures 6–10 we see that the penalized positivity scheme and the square root scheme make almost no difference in the resulting objective function value, and in the case of Poblano, none at all. We see that ASP results in very large objective function values, that sometimes increase, which tells us that is package is not suitable for this particular application. The other packages fall into a smaller range close to each other, and the smallest objective function values are a result of the LBFGS and LBFGSB algorithms which can be observed in Figure 11. Figure 11 combines the resulting objective function values for varying packages and algorithms. Not all tested methods are plotted as the resulting plot is too difficult to read. Since several variations of algorithms have been tested, the depicted plots are those that resulted in lower objective function values.

Reconstructions

Figures 12–13 are the observation and pristine image, while Figures 14–21 are selected reconstructions. The selected reconstructions do not include all reconstructions, but are intended to give a sampling of the different algorithms and packages.

The reconstructions shown contain a mix of the two positivity constraints. Just as the plots of the objective functions show near-identical values, there is little difference in the two reconstructions when comparing the same algorithms with different positivity enforcement schemes. When put side by side virtually no distinction can be made. The presented

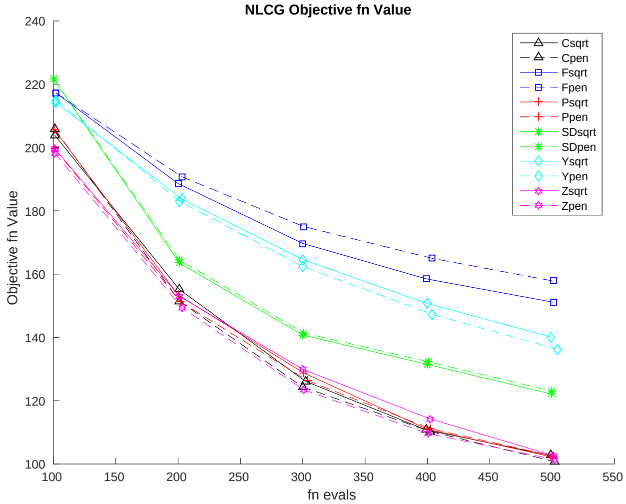

Figure 7. F is Fletcher-Reeves. P is Polak-Ribiere-Polyak. C is P constrained by F. SD is steepest descent. Y is Dai-Yuan. Z is Hager-Zhang.

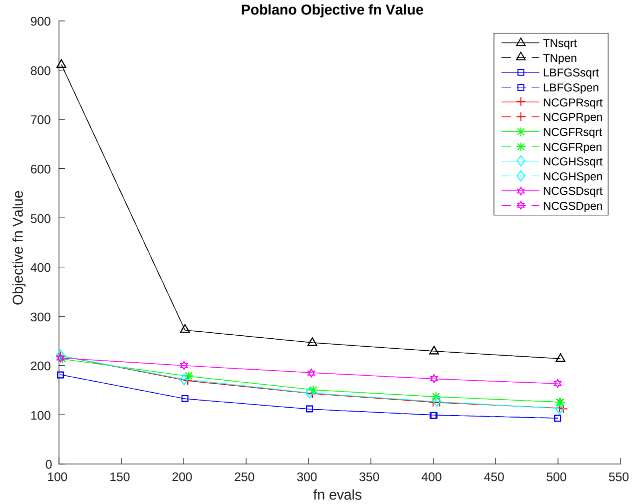

Figure 9. TN is Truncated Newton. NCGPR is Polak-Ribiere-Polyak. NCGFR is Fletcher-Reeves. NCGHS is Hestenes-Stiefel. NCGSD is steepest descent.

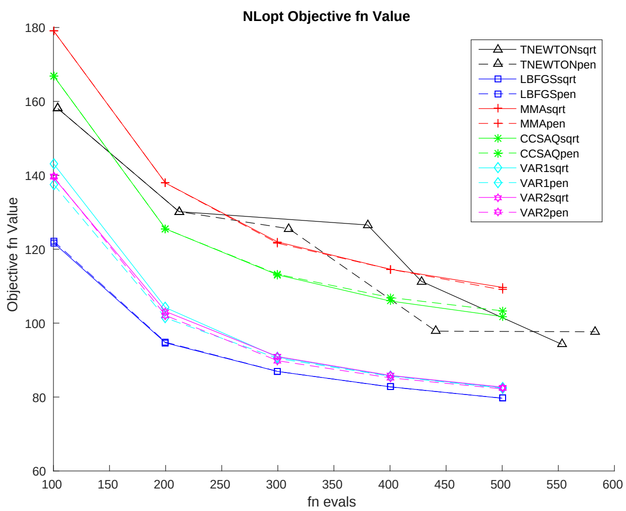

Figure 8. The positivity scheme only appears to have minor effects, the most noticable being on the truncated Newton method

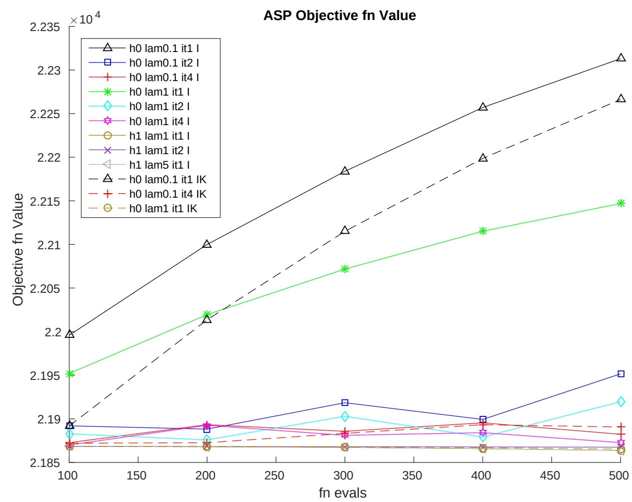

Figure 10. “h0” refers to the homotopy option turned off. “h1” refers to the homotopy option turned on. “lam” refers to the regularization parameter λ used. “it” refers to the number of iterations the function evaluations were split over. “I” refers to only image function evaluations counted. “IK” refers to both image and kernel function evaluations counted.

estimates are also those at the 500 function evaluation level since those all appeared to be more detailed and not yet over processed.

Subjective assessment of the reconstructed images is consistent with the trends observed in the objective function values, with the quasi-Newton methods delivering the best performance. The LBFGSB reconstruction shown in Figure 19 contains more satellite detail than any other method tested. The NLopt implementations of LBFGS and SLMVM (see Figure 17) are only marginally worse. The Poblano LBFGS (Figure 18) is not as detailed.

Methods that are able to recover some image detail but not as much as the above methods are: NLCG Polak-Ribiere-Polyak constrained by Fletcher-Reeves, NLCG Polak-Ribiere-Polyak, NLCG Hager-Zhang, Poblano Polak-Ribiere-Polyak, Poblano Hestenes-Stiefel, Poblano LBFGS, and NLOPT truncated Newton.

Methods that reconstructed with little to no improvement to the average of the input image frames are: NLCG Fletcher-Reeves, NLCG steepest descent, NLCG Dai-Yuan, Poblano Fletcher-Reeves, Poblano truncated Newton, and Poblano steepest descent.

Methods that resulted in reconstructions that were visibly worse than the average of observations are: NLOPT MMA, NLOPT CCSAQ, and ASP.

5. Future Work

To make a “fair” comparison between the optimization methods in this paper, we stopped each algorithm after a fixed number of function evaluations. However, this does not take into account that some of the algorithms (e.g. nonlinear

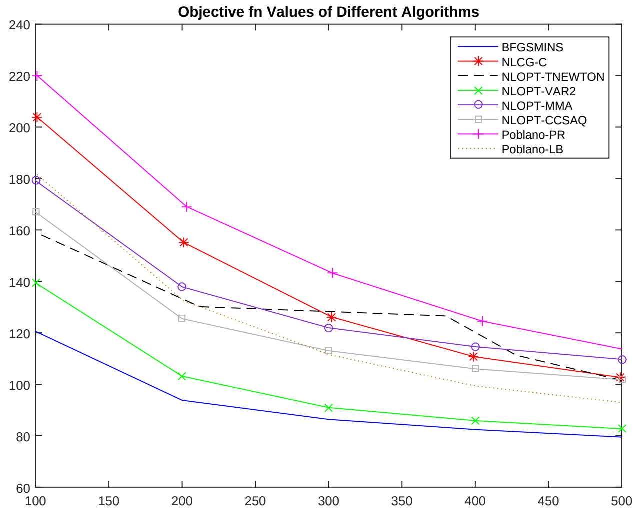

Figure 11. A variety of objective function value results from different packages and algorithms. Each method utilizes the square root scheme for positivity constraint.

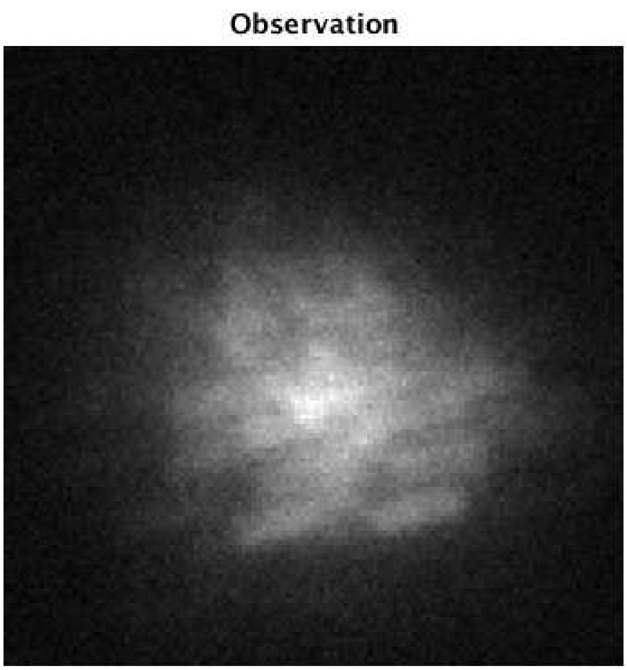

Figure 12. This is one of 30 frames in our observation.

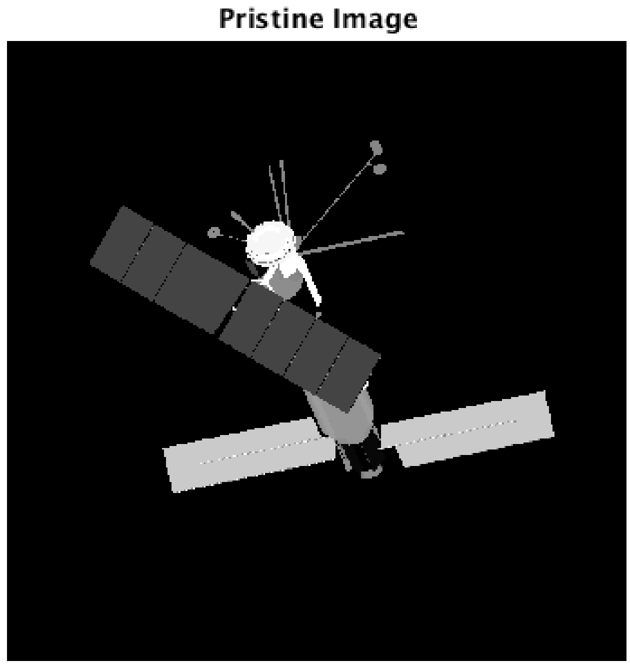

Figure 13. This is the pristine image used to generate our observations.

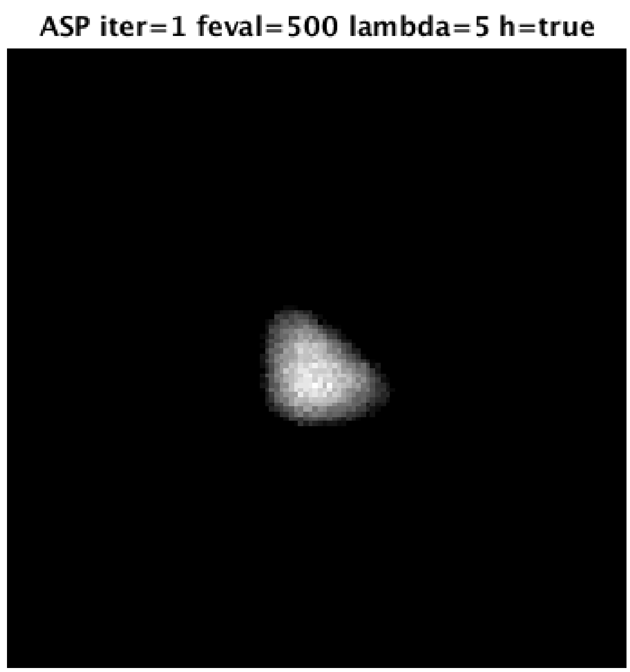

Figure 14. This reconstruction enforces the "homotopy" option. The other ASP reconstructions look similar to this, so we can see that ASP is not suited to this particular application.

conjugate gradient) can be efficiently parallelized, while others (e.g. LBFGS) cannot due to communication costs. In Figures 22–26 we show that with more evaluations the NLCG package can produce reconstructions similar to LBFGSB. In future work, we will expand our comparison to include a degree of parallelization.

6. Conclusions

This paper compares the effectivenes of numerous algorithms and packages for the purpose of minimizing the MFBD objective function (1). The algorithms with the lowest resulting objective function are LBFGSB and LBFGS, followed closely by the NCG methods Hager-Zhang, Polak-Ribiere-Polyak, and Polak-Ribiere-Polyak constrained by Fletcher-

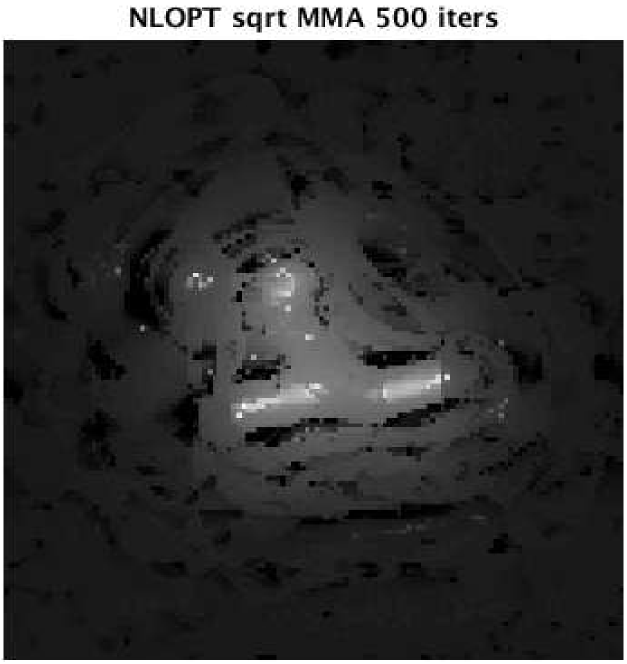

Figure 15. CCSA leads to a similar reconstruction as MMA. Although both yield low objective function values the resulting reconstruction is not close to the true image.

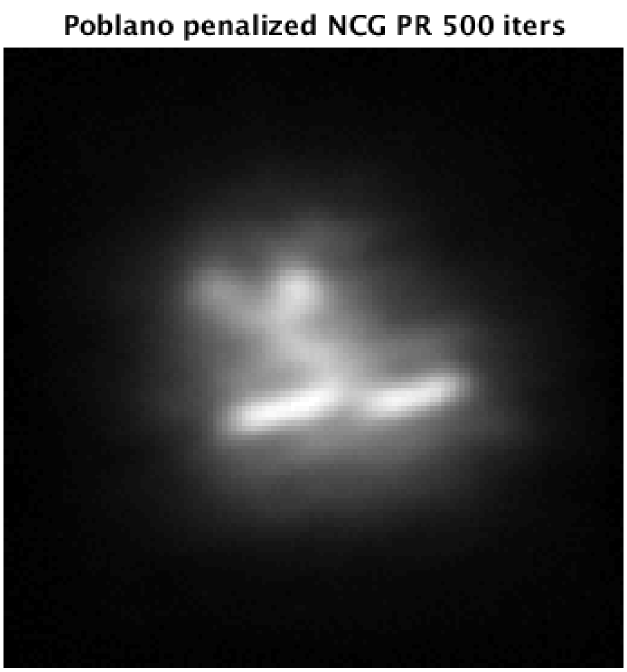

Figure 16. The NCG methods from Poblano and NLCG look very similar, but there are small differences. This reconstruction contains more satellite details than other Poblano NCG reconstructions.

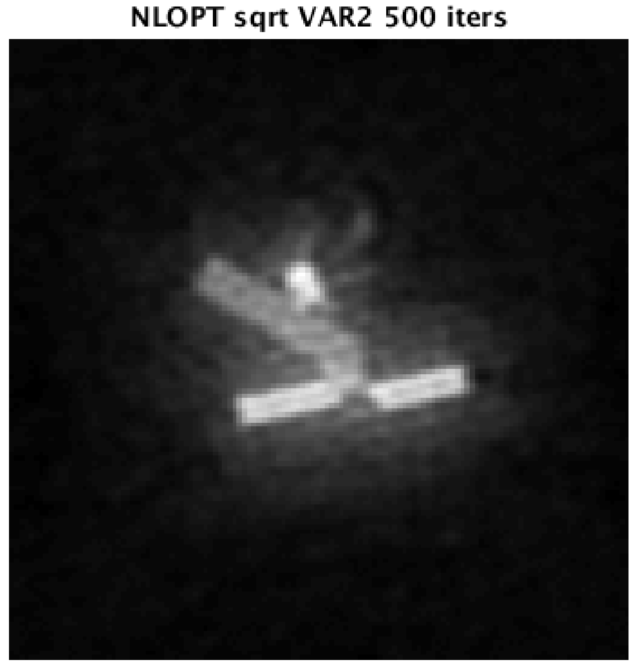

Figure 17. Var2 refers to the rank 2 update. The resulting rank 1 estimate is nearly indistinguishable from the rank 2. This method is similar to BFGS methods and results in similar reconstructions. The NLOPT LBFGS method also appears similar to this.

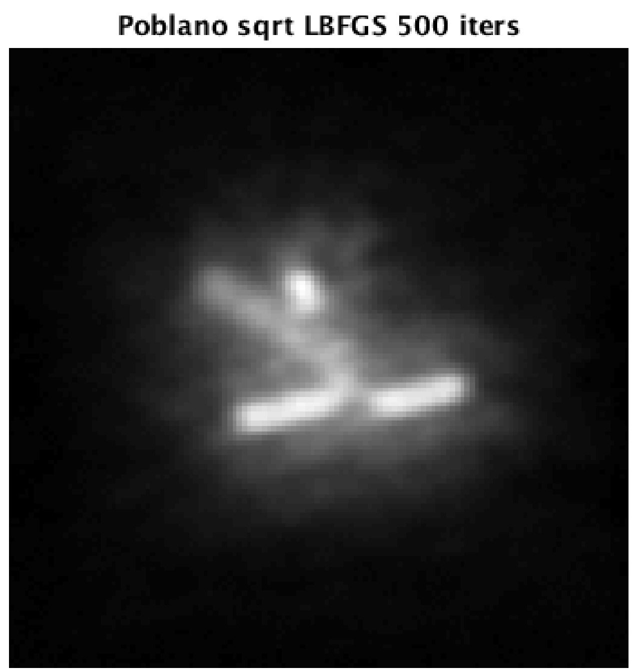

Figure 18. BFGS based reconstruction contained the most details in all our estimates, and this is the most detailed from Poblano. However, it does not contain as many details as the other BFGS based methods.

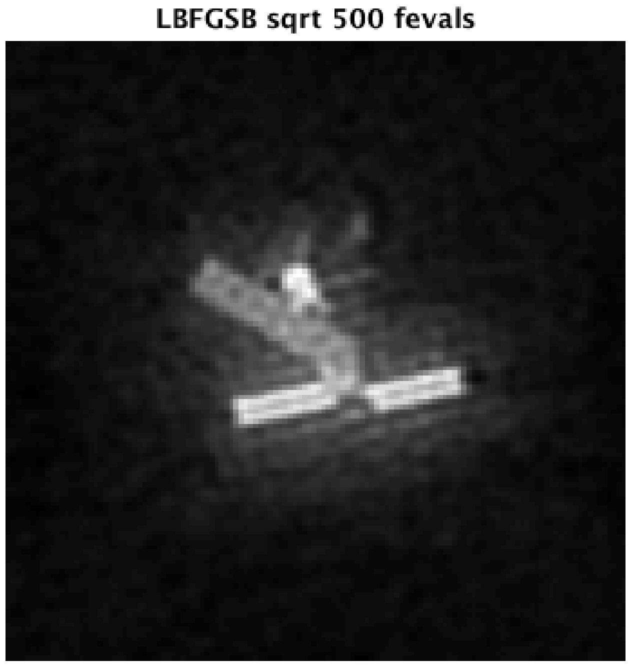

Figure 19. This reconstruction appears to have the most details and sharpest edges.

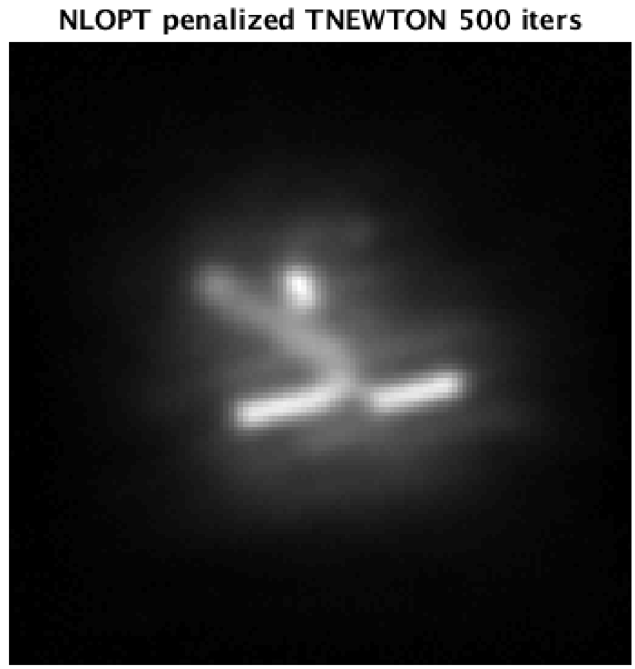

Figure 21. The NLOPT truncated Newton method resulted in a reconstruction with more satellite details than the Poblano version.

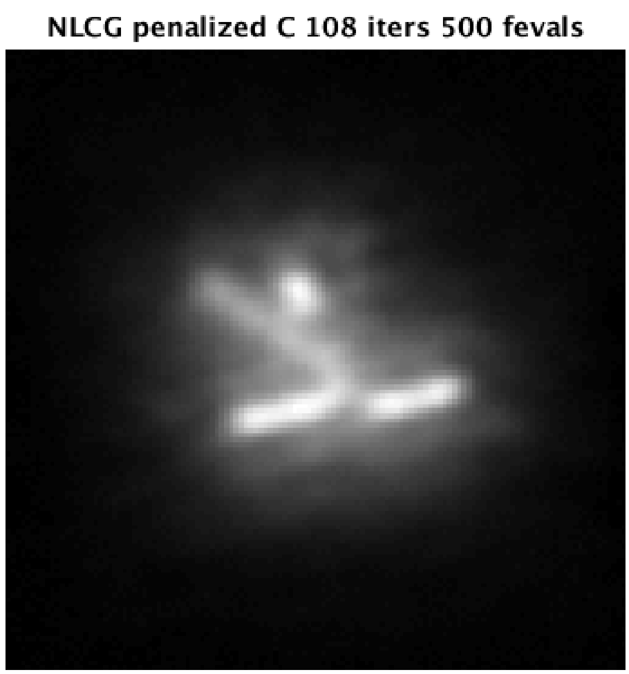

Figure 20. This is the Polak-Ribiere-Polyak constrained by Fletcher-Reeves reconstruction. It is visually similar to the Dai-Yuan and Hager-Zhang reconstructions. The other NLCG and NCG results do not have as much detail.

Figure 22. This is the Polak-Ribiere-Polyak constrained by Fletcher-Reeves case. After 1100 function evaluations this method appears almost as detailed as LBFGSB case at 500 function evaluations.

Figure 23. This is the Polak-Ribiere-Polyak constrained by Fletcher-Reeves case. After 2200 function evaluations this method appears to be more detailed than the LBFGSB case at 500 function evaluations.

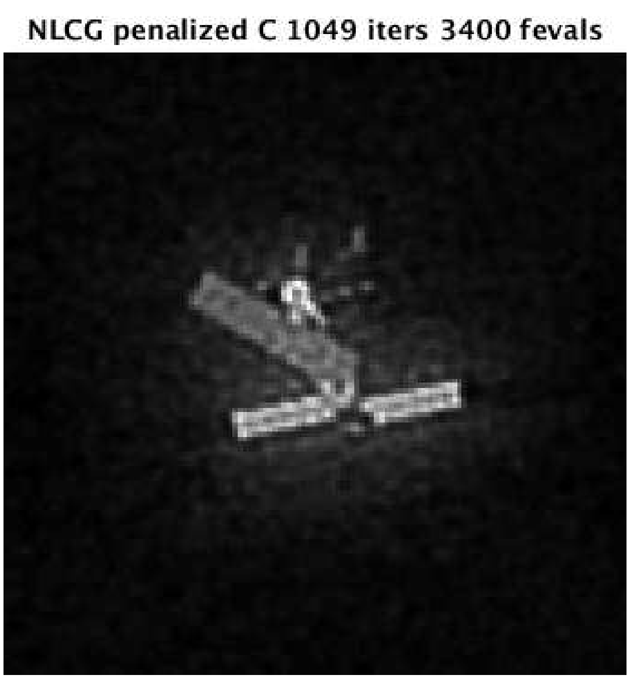

Figure 24. This is the Polak-Ribiere-Polyak constrained by Fletcher-Reeves case. There is some increase in detail here but overprocessing may soon be a concern with more function evaluations.

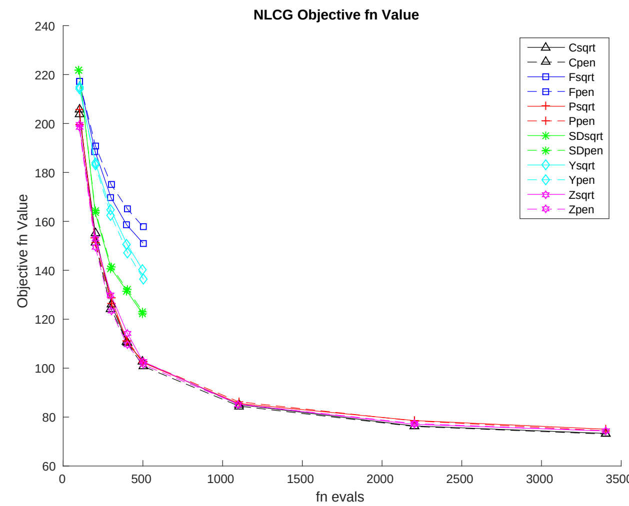

Figure 25. F is Fletcher-Reeves. P is Polak-Ribiere-Polyak. C is P constrained by F. SD is steepest descent. Y is Dai-Yuan. Z is Hager-Zhang

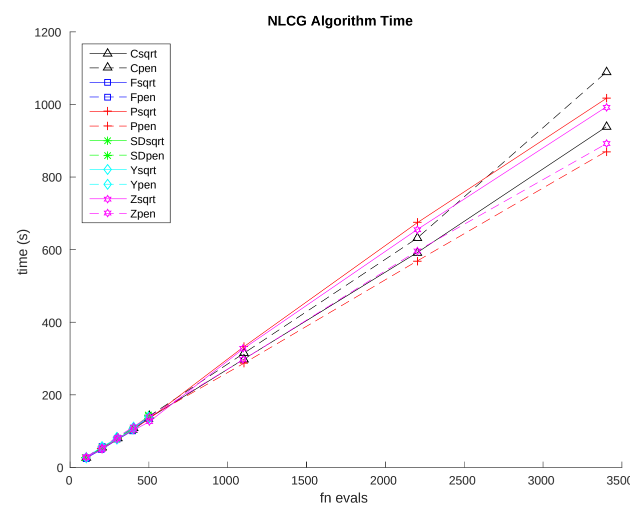

Figure 26. F is Fletcher-Reeves. P is Polak-Ribiere-Polyak. C is P constrained by F. SD is steepest descent. Y is Dai-Yuan. Z is Hager-Zhang. Increased function evaluations follows a linear increase in time.

Reeves.

The most subjectively appealing reconstructions from simulated test data were obtained by LBFGSB, NLOPT LBFGS and NLOPT SLMVM, although given enough evaluations, the NLCG package yields similar performance. Since the package is also easily parallelizable compared to the other methods, it would interesting to see its performance compared to other methods that are not easily parallelizable such as LBFGSB.

ACKNOWLEDGMENTS

This work was funded by the Air Force Office of Scientific Research through AFRL contract FA9451-05-C-0257.

REFERENCES

[1] T. J. Schulz, "Multiframe blind deconvolution of astronomical images," Journal of the Optical Society of America, vol. 10, no. 5, pp. 1064–1073, 1993.

[2] C. L. Matson, K. Borelli, S. Jefferies, J. Charles C. Beckner and E. K. Hege, "A fast and optimal multiframe blind deconvolution algorithm for high-resolution ground-based imaging of space objects," Applied Optics, vol. 48, no. 1, pp. A75–A92, 2009.

[3] C. Zhu, R. H. Byrd, and J. Nocedal, "L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization," ACM Transactions on Mathematical Software, vol. 23, no. 4, pp. 550–560, 1997.

[4] S. G. Johnson, "NLopt," Massachusetts Institute of Technology, 3 October 2013. [Online]. Available: http://ab-initio.mit.edu/nlopt.

[5] M. L. Overton, "NLCG: Nonlinear Conjugate Gradient," New York University, 27 August 2015. [Online]. Available: http://www.cs.nyu.edu/faculty/overton/software/nlcg/index.html.

[6] D. M. Dunlavy, T. G. Kolda, and E. P. Acar, "A MATLAB Toolbox for Gradient-Based Optimization, Technical Report SAND2010-1422," Sandia National Laboratories, Albuquerque, NM and Livermore, CA, March 2010.

[7] M. P. Friedlander and M. Saunders, "ASP," University of British Columbia, 26 April 2015. [Online] Available: http://www.math.ucdavis.edu/~mpf/asp/.

[8] M. Friedlander and M. Saunders, "A dual active-set quadratic programming method for finding sparse least-squares solutions, DRAFT Technical Report," Dept of Computer Science, University of British Columbia, Vancouver, 2012.

[9] R. Dembo and T. Steihaug, "Truncated-Newton algorithms for large-scale unconstrained optimization," Mathematical Programming vol. 26, pp. 190–212, 1983.

[10] R. Dembo, S. Eisenstat, and T. Steihaug, "Inexact Newton Methods," SIAM J. Numer. Anal. vol. 19, pp. 400–408 (1982).

[11] L. Luksan, "Ladislav Luksan," Academy of Sciences of the Czech Republic, 8 April 2015. [Online]. Available: http://www.uivt.cas.cz/~luksan/subroutines.html.

[12] J. J. Moré and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease," ACM Trans. Math. Software, no. 20, pp. 286–307, 1994.

[13] J. E. Dennis and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Philadelphia: SIAM, 1996.

[14] J. Nocedal and S. J. Wright, Numerical Optimization, Springer, 1999.

[15] J. Nocedal, "Updating quasi-Newton matrices with limited storage," Mathematics of Computation, no. 35, pp. 773–782, 1980.

[16] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Mathematical Programming, no. 45, pp. 503–528, 1989.

[17] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," SIAM Journal on Optimization, vol. 12, no. 2, pp. 55–573, 2002.

[18] J. Vlcek and L. Luksan, "Shifted limited-memory variable metric methods for large-scale unconstrained minimization," Journal of Computational and Applied Mathematics, no. 186, pp. 365–390, 2006.

[19] R. Fletcher and C. Reeves, "Function minimization by conjugate gradients," The Computer Journal, vol. 7, no. 2, pp. 149–154, 1964.

[20] E. Polak and G. Ribiere, "Note sur la convergence de methods de directions conjugres,," Revue Francaise Informat. Recherche Operationnelle, vol. 16, pp. 35–43, 1969.

[21] W. W. Hager and H. Zhang, "A New Conjugate Gradient Method With Guaranteed Descent and an Efficient Line Search," SIAM Journal on Optimization, vol. 16, no. 1, pp. 170–192, 2005.

[22] Y. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," SIAM Journal on Optimization, vol. 10, no. 1, pp. 177–182, 1999.

[23] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," Journal of Research of the National Bureau of Standards, Section B, vol. 48, pp. 409–436, 1952.

[24] P. Wolfe, "Convergence Conditions for Ascent Methods," SIAM Review, vol. 11, no. 2, pp. 226–235, 1969.

[25] P. Wolfe, "Convergence Conditions for Ascent Methods. II: Some Corrections," SIAM Review, vol. 13, no. 2, pp. 185–188, 1971.

[26] P. Gill and W. Murray, "Safeguarded steplength algorithm for optimization using descent," Technical Report NPL NA 37, 1974.

BIOGRAPHY

Daniel Thompson *received a BA in mathematics from Willamette University and an MS in Applied Mathematics from University of California Merced in 2011 where most of his research has dealt with image reconstruction. His research interests include image reconstruction, optimization, and high-performance computing. Currently Daniel is an imaging scientist with Boeing.*

Brandoch Calef *is an imaging scientist with Boeing. For ten years, he has supported the AFRL space surveillance site on Maui. His research interests include image reconstruction, atmospheric statistics, estimation theory, optical instrumentation, and high-performance computing. He received a PhD in Applied Mathematics from the University of California Berkeley.*

Michael Werth *is a system engineer and imaging scientist at Boeing. He primarily works on developing new data analysis algorithms that leverage supercomputing resources and new optical sensors at the Air Force Maui Optical and Supercomputing site. He received a Ph.D. in Physics from UC Irvine in 2012 after a graduate career spent working with the Large Hadron Collider.*