An Automated HPC Implementation of the Metric Information Network (MIN)

C. Johnston, C. Bleiler, O. Sjahputera, M. Karspeck, B. Bader, R. Ravi

The Metric Information Network (MIN) system for generating large-scale ground control networks from satellite imagery was introduced by its developers in 2007 (see citation [DI07]). The current paper describes a large-scale, high performance computing (HPC) implementation of the MIN methodology that is under development at DigitalGlobe. We describe the DigitalGlobe system's capabilities, and give experimental results from our replication of a MIN experiment over California and Nevada with 50 stereo pairs of DigitalGlobe WorldView-1 imagery. The original experiment at BAE was described in citations [DS10A-B].

1. INTRODUCTION

In [DI07], researchers at BAE introduced the Metric Information Network (MIN), a system for the generation of an extensive ground control point network from aerial or satellite imagery. The term 'MIN' is used to describe both the system for generating the ground control point (GCP) network, and the GCP network itself.

A MIN has high absolute accuracy and is internally consistent, i.e., has high relative accuracy. It is constructed sequentially from a series of typical block adjustments, each using tie points extracted from statistically independent sets of input imagery. Each block adjustment is followed by a fusion step (the Stage 2 MIN update, see [DI07]) that incorporates the newly added information consistently into the existing MIN. The resulting geodetic network is (roughly) equivalent to the GCP network that would be constructed from a batch bundle adjustment using equivalent inputs; however, a MIN is built incrementally, and may cover an extremely large area.

A MIN may utilize images from multiple sensors for its input observations. Each additional independent observation of a given GCP reduces the statistical error in the associated object point location. Externally surveyed GCPs can be incorporated into a MIN in a consistent way, thereby improving the absolute accuracy of the MIN.

MIN GCPs have correlated errors, as do object points resulting from a batch bundle adjustment, and their posterior cross-covariances can be derived as a by-product of the MIN process. Cross-covariances between points are stored in the MIN database, and are used in every Stage 2 update to maintain the high relative accuracy of the GCP network. Tie point image chips (i.e., smaller subimages centered on GCP observations) derived from input images are also stored in the MIN database, and can be used to identify GCPs in new imagery either manually or automatically.

Orthomosaic production is an important application of the MIN technology. Large orthomosaics produced from satellite or aerial images often contain noticeable misalignment errors at the edges of input scenes, even when the orthomosaic meets absolute accuracy specifications. Misalignment between scenes can be reduced using a MIN defined over the entire region of the orthomosaic.

As a first step in the MIN process, MIN GCP observations are identified in the input imagery to the orthomosaic, using automated or manual tie point matching between the MIN GCP chips and the input images. The input images are then block adjusted, using the MIN observations, and the adjusted imagery is used to build the orthomosaic. The orthomosaic will then display the higher relative accuracy of the GCP network across image boundaries. Together with automated technology for image source selection, the MIN enables the automatic production of an orthomosaic that is geographically seamless.

We have built an HPC MIN implementation that is designed to take advantage of DigitalGlobe's large archive of WorldView-1 and WorldView-2 satellite imagery. The WorldView instruments are pushbroom electro-optical (EO) sensors that have high pointing accuracies of 3-4 meters on the ground. Panchromatic band ground sample distances (available in commercial imagery) are as small as 0.5 m for the WorldView sensors. More details about the WorldView satellites are available in [DG07, DG09].

HPC MIN OVERVIEW

The mathematics of the MIN fusion algorithm is described in detail in Sections 2-4 of [DI07]. Dolloff et al take a Kalman filter approach to the development of the MIN algorithm; the MIN algorithm can also be derived as a special case of the sequential least squares algorithm [MA82].

Each MIN update consists of the following two stages:

1. **Stage 1.** A standard non-linear least squares block adjustment of the new image set. Image observations in the new image set may include observations of new tie points, and/or observations of existing MIN GCPs in the new imagery. New tie points are treated in the usual way by the block triangulation, and new observations of MIN GCPs are incorporated, together with the MIN GCP prior coordinates and covariances. It is new observations of existing MIN points that tie the new points into the existing MIN network, and new tie point observations that extend the geodetic network.
2. **Stage 2.** A 'fusion' step, in which the information from the new block adjustment is correctly incorporated into the existing MIN. The fusion algorithm utilizes the GCP posterior covariances for points involved in the step 1 block adjustment to correctly adjust the coordinates and covariances for the MIN GCPs that were not included in the block adjustment, resulting in the maintenance of a consistent geodetic network after every MIN update.

Before each Stage 1 block adjustment, observations of new tie points and existing MIN GCPs must be identified in the new input imagery. In the BAE MIN prototype, this identification was human-driven using the Socet Set Adaptive Tie-point Matcher (ATM) program, together with operator review and potential adjustment. ATM for both tie point identification and GCP chip matching was run on new images before each Stage 1 update, so that the image processing component of MIN processing was performed in-line with all the MIN updates.

If a great deal of imagery is available to be processed up front (as is the case with the DigitalGlobe WorldView archive), then the entire image processing component of MIN processing can be completely decoupled from the MIN update component. This is an advantage because tie point identification can be a data-parallel process, whereas MIN updates are necessarily sequential.

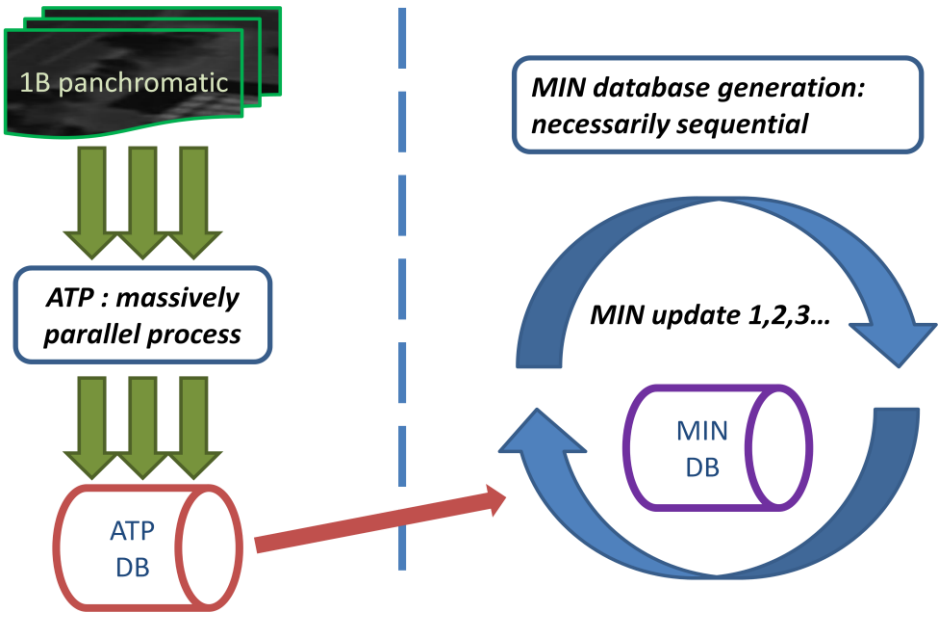

Figure 1: High level architecture of HPC MIN.

Figure 1 shows the high-level architecture of the HPC MIN system. The image processing pipeline of the HPC MIN is called the Automatic Tie pointing Process, or ATP. The ATP extracts image observations of MIN GCPs from

large collections of images, and saves the observations (together with data about which images are used, and which object points are referenced) to a database, the ATP DB. The sequential MIN update process draws on the ATP DB for image point observations and image information as the MIN updates are performed.

The ATP extraction process is "embarrassingly parallel". After interest points are identified in individual images, their planimetric locations are known to within meters, so that potentially matching interest points can be analyzed and grouped in smaller subprocesses confined to quadkeys. The term 'quadkey' refers to level 14 tiles in the Bing Maps Tiling System, each of which has an area of roughly 4 km^2.

Restriction of the processing pipeline to quadkeys works because it is highly likely that a point whose estimated location is in the quadkey really belongs in the quadkey; the set of exceptions to this rule is small enough not to matter. These subprocesses are independent, and can be spread across a cluster without the need for communication between nodes.

The rest of the paper is laid out as follows. Section 2 describes the ATP tie point extraction subsystem. Section 3 describes the MIN update process. Section 4 describes our Edwards AFB study, which replicates a BAE MIN study published in references [DS10A-B].

2. THE ATP SUBSYSTEM

The ATP subsystem populates the ATP database with high-quality image point observations – high quality, in this case, means that the image points are reproducible, i.e., distinctive and persistent. The MIN update process then draws on the pre-populated ATP database for the image points that will drive the numerical MIN update process.

The separation of the ATP and MIN update subsystems achieves two goals. First, it effectively divides the MIN system into an offline image preprocessing portion and a numerically intensive MIN update system. Second, it allows the image preprocessing portion to run as a massively parallel process.

The ATP subsystem consists of a pipeline of processes, in the following order:

1. Orthorectification of all images
2. Interest point/descriptor extraction
3. Interest point grouping into tie point sets
4. Tie point selection
5. Tie point deblundering
6. Tie point optimization
7. Blunder correction and tie point augmentation
8. Further tie point subselection to ATP DB

What follows are high-level descriptions of each ATP subprocess, since the details of ATP processing constitute a paper of their own.

1. **Orthorectification of all images (scene-based)**. The constituent images are all orthorectified to a digital elevation model (DEM), usually SRTM-90, before any further processing is done, using standard methods. This enables interest points to be detected and matched in spite of non-rotational distortions due to different viewing angles in the source 1B images.

2. **Interest point and descriptor extraction (scene-based)**. The orthorectified images are each submitted to a program that detects interest points (i.e., potential ground point observations) and extracts their feature vectors (descriptors), using standard methods. We are also studying the potential for using domain-specific features and filters to improve interest point detection.

3. **Interest point grouping into tie point sets (quadkey-based)**. Interest points in different images are grouped into sets of observations of the same ground point, in a two-stage process. The first step leverages the high geolocation accuracy of the WorldView sensors to do geolocation-based clustering, and the second step leverages interest point descriptors from the previous set.

4. **Tie point selection (quadkey-based).** We will discard most of the TP sets found in the previous step; we will keep only the TP sets that are of highest quality. TP sets are considered to be of high quality if the interest points within them are highly similar, and if they contain a higher number of interest points, as the latter condition implies that the ground point observations are reproducible across images. In this step, TP sets are ranked twice, according to their geolocation scatter and their feature vector scatter (each normalized by cluster size). These ranks are summed into a single rank, and a predetermined number of the top-ranked TP sets are kept for further processing; the rest are discarded.

5. **Tie point deblundering (quadkey-based)**. The remaining TP sets are then reviewed for blunders in individual scenes. The (fully automated) review detects blunder rays in TP sets using two criteria. The first blunder detection method is based on calculating the ground residuals of each interest point. The interest point ground residual is calculated as the distance between the tie point ground location, estimated using standard triangulation methods, and the image-to-ground projection of the interest point (utilizing a DEM). Blunder detection is based on the ground residual passing both a user-based threshold test, and a robust distribution-based threshold test. The second blunder detection method utilizes a 'correlation fitness' metric, applied to each ray individually, which measures its normalized cross-correlation relative to the other interest points in the group. A stringent threshold is applied for acceptance of an interest point in this test. These blunder detection methods are applied iteratively to the interest points in the tie point set. The iterations terminate when either no more blunders are found, or only two rays remain.

6. **Tie point optimization (quadkey-based).** In this step, after deblundering, we adjust the locations of each interest point in a tie point, in order to maximize the pairwise correlation measures for each interest point pair. All interest points within a tie point are treated equally, and their locations are adjusted simultaneously in order to maximize the aggregate correlation measure.

7. **Blunder correction and tie point augmentation (quadkey-based).** Having identified blunder rays in Step 5, removed them from the tie point set, and optimized the remaining interest points, we attempt to apply corrective adjustments to the removed blunder rays and readmit them to the tie point set. This is done by perturbing the location of the blunder ray in order to reduce its aggregated normalized cross-correlation peak, relative to the other interest points. Finally, we attempt to augment the original tie point set, by detecting new observations of the same feature in any images whose footprints overlap the area of the existing tie point, but which did not contribute interest points in the original tie point set. This can occur for a number of reasons, some intrinsic to the image (snow or clouds over the point, for example), and some due to processing (i.e., an interest point's being assigned to a different cluster). In this step, we first obtain an interest point candidate in the extra image by calculating ground-to-image on the triangulated estimate of the ground point location for the tie point. The blunder correction algorithm is then applied to the interest point candidate.

8. **Tie point subselection to an ATP DB.** The final selection of tie points to be used in the MIN depends on the desired density of tie points. At this stage, we downsample the tie points in the interim database so that we are keeping only 1 tie point per 4 quadkeys (1 per 16 $km^{2)}$. We simply select a tie point at random, from the set of tie points with the most rays.

3. IMPLEMENTATION OF THE MIN UPDATE SUBSYSTEM

The MIN update algorithm was described in detail in [DI07]. In the DG MIN implentation, all its inputs are drawn from the ATP and MIN DBs; no image processing is done during a MIN update, as it is all confined to the ATP subsystem. Each MIN update is defined by specifying which images are to be added to the MIN inputs during the update.

Stage 1. All image point observations associated with the specified imagery are already resident in the ATP database, and all are pulled for inclusion in the Stage 1 block adjustment input. This will include additional observations of object points already represented in the MIN DB (called 'group 1a' points), and observations of new object points not yet introduced to the MIN DB ('group 1b' points). Object points in the MIN that are not observed in the current image set are called 'group 2' points. As the MIN grows larger, group 2 points will outnumber group 1a and 1b points on every update (see Figure 2 for a diagram of these point sets).

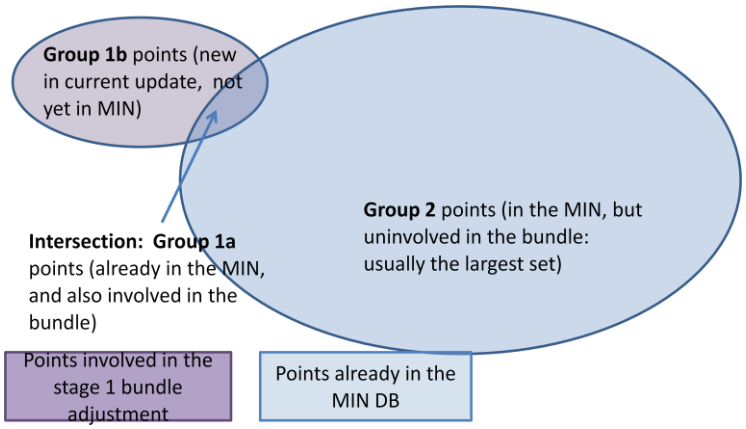

Figure 2: Schematic of object points sets involved in a MIN update.

If an image is to be included in an update, all other correlated images must be included as well; therefore, image sets for MIN updates are unions of complete correlated image sets. As a rule of thumb, WorldView-1 images from the same revolution are considered to have correlated errors, and a priori cross-covariances between correlated image parameters are modeled in the bundle adjustment. Cross-covariances between parameters from different images decay with the time separation between images, and are modeled so that the resulting a priori image covariance matrix remains positive definite [DLST06,D10].

The Stage 1 block adjustment is a standard nonlinear least squares solver using the Levenberg-Marquardt nonlinear optimization algorithm. The DigitalGlobe Community Sensor Model (CSM) for WV01 is integrated into DigitalGlobe's bundle adjustment implementation, and provides the WorldView sensor models (i.e., image-to-ground, ground-to-image, and derivative calculations) [CSM07]. Note that the WV01 CSM, which was developed at DigitalGlobe, is distinct from the sensor model for WV01 used in BAE's Socet Set software. Our CSM-based implementation of bundle adjustment takes advantage of the sparsity of the normal matrix, and uses the reduced camera system approach to solving the normal equation, in a manner similar to that implemented in the open source SBA package [LA09]. However, the SBA assumes that all object points and camera parameters are *a priori* uncorrelated. This is a typical assumption in standard block adjustment problems, and it results in a very sparse normal matrix (see Figure 3, left).

The MIN algorithm, however, requires that points included in the Stage 1 block adjustment be modeled with their cross-covariances. In addition, the block adjustment must also model image pairs having correlated errors. Correlated sets of parameters, whether they are object point or image parameters, give rise to larger diagonal blocks in the normal matrix for the Stage 1 block adjustment (see Figure 3, right). The MIN's block adjustment implementation therefore must adapt to the problem. In cases where there are no correlated parameter sets in the input data, the algorithm is equivalent to the SBA algorithm and therefore is very efficient; but it can also treat larger blocks in the normal matrix as block components of a sparse problem.

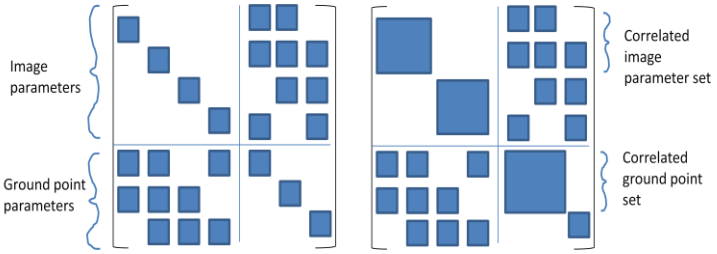

Figure 3: Structure of normal matrices for (l) classic block adjustment problem with uncorrelated parameters; (r) block adjustment with a priori correlated parameter sets.

There were some differences between the DG and Socet Set bundle adjustments for this experiment. BAE allowed adjustments in 6 sensor model parameters, 0^{th}- and 1^{st}-order roll, pitch and yaw; DG allowed adjustments in 5 sensor model parameters, 0^{th}-order roll, pitch, and yaw, and 1^{st}-order roll and pitch. Socet Set's default prior standard deviations for all 3 0^{th}-order attitude parameters were 5 μrads; the DG bundle adjustment used 2 μrads for 0^{th}-order roll and pitch, and 4 μrads for yaw.

The Stage 1 process is a full photogrammetric block adjustment, in that it outputs not only the modified point coordinates, but also portions of the posterior covariance matrix of the block adjustment. Specifically, all image-to-image and point-to-point components of the posterior covariance matrix are calculated; the image-to-point components of the posterior covariance matrix are expensive to calculate and are only needed for diagnostic purposes, so they are not routinely calculated. The posterior point-to-point covariance blocks from the block adjustment are included in the MIN DB as part of the current MIN update; the Stage 2 step is needed in order to keep the rest of the MIN point coordinates, and their covariances, consistent with the new MIN update.

Stage 2. After the Stage 1 MIN update, the posterior coordinates and cross-covariance blocks are calculated for point groups 1a and 1b. However, coordinates and covariances of points in group 2 are also affected by the information captured in the new MIN update (refer to Figure 2 for definitions of the point groups). Coordinates of points in group 2 will change slightly, and cross-covariance blocks 1a-2, 1b-2, and 2-2 must be updated (or in the case of 1b-2, initially defined). This update involves computing the inverse of the 1a-1a block of the posterior covariance matrix from the Stage 1 MIN update, and performing a series of large matrix multiplications involving the components of the posterior and prior covariance matrices. Mathematical details of the MIN Stage 2 algorithm may be found in [DI07].

The computational expense of MIN Stage 2 changes with the sizes of the various point groups. As the MIN grows in size after multiple block updates, the largest, most expensive calculation in the MIN update will eventually be the posterior covariance matrix of block 2-2, P_{22}. The calculation time is $O(m^2n)$, where m is the number of group 2 points and n is the number of 1a points. Storage of this matrix is $O(m^2)$, so file I/O involving this matrix can be a limiting factor. An experimental MIN over Denver, Colorado north to Cheyenne, Wyoming with only 8300 points, for example, produced a 4.2 GB binary covariance matrix file (even with the space savings due to storing only the upper triangular portions of the symmetric matrix).

Most of the computational burden of the MIN, therefore, comes from the need to store and manipulate large covariance matrix blocks when the MIN grows large. Our Stage 2 algorithm therefore makes use of standard methods for distributed, large-scale matrix multiplication using MPI.

Our first implementation of the Stage 2 fusion algorithm was heavily multithreaded on a single machine and therefore limited by the amount of RAM on the node. Even on machines with 256 GB RAM, our potential MIN sizes were limited to approximately 50,000 points. We therefore have implemented a distributed parallel version of the Stage 2 algorithm, which spreads the matrix multiplication across multiple nodes in a cluster. The P_{21a} matrix is distributed row-wise, and blocks of P_{22}, grouped by rows, are calculated on separate compute nodes. This simple parallel scheme avoids the main bottleneck of storing the largest matrix on a single node, and permits larger MINs to be generated.

4. EDWARDS AFB MIN REPLICATION STUDY

As a test of the HPC MIN processing system, we replicated the study described in [DS10A-B]. This study was an assessment of a MIN built using 50 overlapping stereo pairs of high-resolution, panchromatic imagery captured by the DigitalGlobe WorldView-1 satellite. This data, along with 107 survey GCPs, was provided to BAE by DigitalGlobe for the purposes of the study (about 90 of the checkpoints were inside the area covered by the image footprints). The imagery covered a roughly 50,000 km^2 area of the Western United States, including the Edwards, Fallon, and Nellis Air Force Bases (for brevity, we will refer to it as the "Edwards AFB study").

The layout of the imagery, together with the locations of all of the DigitalGlobe-provided GCPs, is shown in Figure 4. The makeup of the image set, which includes East-West acquisitions over two geocells also covered by North-South strips and over a bridging strip to the Nellis geocell, is described in detail in [DS10A].

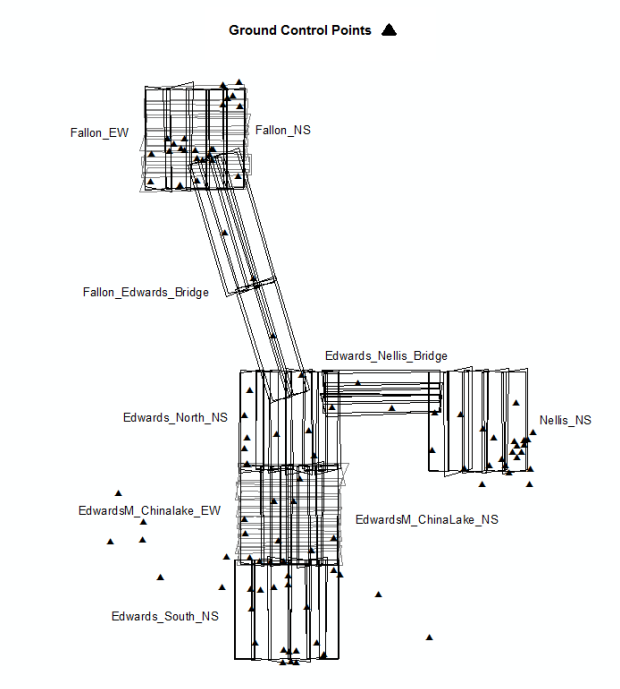

Figure 4: Layout of 50 stereo pairs used for the MIN replication study, with 107 checkpoints available at DigitalGlobe. 50 of these checkpoints were utilized in the study.

In its study, BAE utilized 50 of the DigitalGlobe-provided GCPs as checkpoints, together with an additional 51 checkpoints from other sources. These additional checkpoints were not available to DigitalGlobe for the current study. We therefore used as checkpoints only the 50 GCPs that were both provided by DigitalGlobe, and utilized by BAE. The distribution of the *a priori* accuracies of these tie points are shown in Table 1. The CE90 ranges are used to bin the points for counting, and the LE90 column shows the ranges of LE90 for points in each bin.

Number of points	CE90 range (m)	LE90 range (m)
23	<=.25	<=.3
14	>.25, <=.5	>.3, <=.8
13	>.5, <=.7	>.6, <=1.2

Table 1: Distribution of prior accuracies of the 50 checkpoints.

A major feature of the DigitalGlobe HPC MIN implementation is that the tie point calculation process is fully automated. The only manual step in image measurement determination is the measurement of survey GCPs into the imagery, using Socet Set, to serve as checkpoints. 4686 tie points, of which 4636 were derived fully automatically using the ATP system, were used in this study. In production, these point sets would be reviewed and possibly modified for quality control; but for the purpose of this study, which was intended to measure the performance of the fully automated system, they were quickly reviewed but not modified.

The ATP extracts one MIN point per level 13 quadkey, or roughly one point per 16 km^2. The quadkey-based extraction method ensures some uniformity of MIN coverage over a study area, although in some especially featureless areas, the ATP may not find a sufficiently high-quality tie point in an entire quadkey, and in that case, the quadkey is left empty. A high level of uniformity and density is desirable for applications such as bundle adjustment and orthorectification, because it allows MIN points to be found in new imagery that is heavily occluded by clouds or affected by changes.

The ATP produces a set of tie points that is much larger, denser, and more uniformly distributed than the original BAE MIN over the same study area. The BAE MIN contained 1246 points, distributed mostly along the overlap areas at the edges of stereo pairs. All BAE tie points were observed by either 2, 4, 6, or 8 rays, with a preponderance of 4-ray points; the DigitalGlobe MIN contained points with ray counts ranging between 1 and 11 (most points were observed by 2-4 rays).

Table 2 shows the distribution of ray counts for MIN points in the BAE and DG studies.

The DG MIN was created in 49 MIN updates. Every stereo pair was included in its own update, except for two stereo pairs in different parts of the study area which were acquired in the same pass; these were processed together in the 49th update. Each image strip (itself composed of 10-20 independent scenes) was modeled as a single image strip, contributing five adjustable parameters to the bundle adjustment; 0-order roll, pitch and yaw, and 1-order roll and pitch parameters were adjusted. Images in a single, same-pass stereo pair were modeled as being 75% correlated (at a time separation of 60 seconds). A simple, exponential time-decay model (described in [D10]) was utilized to ensure that the bundle adjustment prior covariance matrices remained positive definite.

RAYS	**BAE**	**DG**
2	472 (35%)	1714 (37%)
3-4	697 (52%)	1539 (33%)
5-6	152 (11%)	916 (19%)
7-8	26 (3%)	462 (10%)
9-12	0 (0%)	55 (1%)
TOTAL	**1347**	**4626**

Table 2: MIN GCP counts from BAE and DG studies, broken down by number of rays (see [DS10A-B]).

Accuracy of MIN results

Table 3 presents extraction accuracy results from our experiment. These results can be compared to those given in [Table 5, DS10B]. As in that study, we examine both the measured error percentile and the predicted accuracy of the resulting MIN points. The measured error percentiles were calculated from the raw magnitudes of the planimetric and vertical errors.

Also included in the table are predicted average CE90 and LE90 metrics. These are the averages of the predicted 90th percentile of planimetric and horizontal error per checkpoint. The per-checkpoint CE90 and LE90 metrics are calculated using the checkpoint's posterior covariance matrix at the end of MIN processing, according to a method used in [DS10A-B] and described in [Appendix F, DTL12]. The 'percentage-of-checkpoints' metrics give the percentage of checkpoints whose errors fell within their per-checkpoint CE90 and LE90. Finally, the percentage of checkpoints having measured error lying within their 3-D 90% error ellipsoid was calculated, and is listed as Spherical error.

	Measured error percentile, (m)			Predicted Accuracy	
	50%	90%	95%	Predicted average LE90/CE90 (over checkpoints)	% of checkpoints within their predicted 90% error (range/ellipse/ellipsoid)
Horizontal	1.22	2.6	2.8	1.06	48%
Vertical	.78	1.4	1.47	1.64	86%
Spherical	-	-	-	-	76%

Table 3: Extraction accuracy metrics.

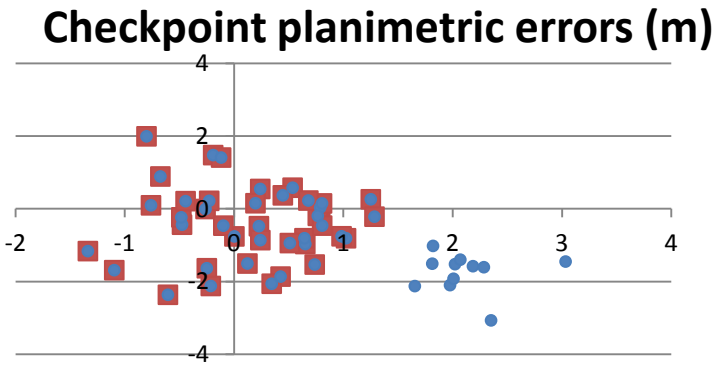

Figure 5: Checkpoint planimetric errors for 50 checkpoints. The smaller blue points to the southeast are the 11 checkpoints associated with the Fallon-NS block.

On a sufficiently large sample set of high-quality GCPs, each of these metrics (particularly the spherical error metric) should approach 0.9. The LE90 metric behaves as expected; however, the measured planimetric error is significantly higher than predicted by the posterior covariances.

The "% of checkpoints within their predicted 90% error range" statistic is an indicator of the internal consistency of the statistical model of the data. If the model (including the assigned temporal correlation between same-pass images) is consistent, then this statistic should be nearly 0.9. The vertical consistency of the model is nearly correct, but the planimetric statistic is low. This pattern was also seen in the BAE study, but was much less extreme there. A measured LE90 statistic that is significantly smaller than the predicted LE90 is consistent with underestimating the temporal correlation between images in a stereo pair, but on a checkpoint sample as small as this one, we cannot claim with certainty that this is occurring.

We suspect, instead, that these statistics are due to a significant southeasterly bias in some of the planimetric errors. These errors arise from the 11 checkpoints in the northernmost (Fallon-N) block (see Figure 5 – the Fallon-NS

checkpoints are the smaller blue dots to the southeast). In Figure 6, one also notes a small data dropout in the MIN in the Fallon block of data where the ATP did not extract tie points. A study of the input orthorectified images in this area shows a great deal of distortion, due to inaccuracies in the terrain data used for orthorectification; the combination of terrain error, and lack of persistent features in the imagery, caused the data dropout in this area. Note that posterior CE90 estimates in the area around the hole are significantly higher than they are elsewhere in the Fallon-N block.

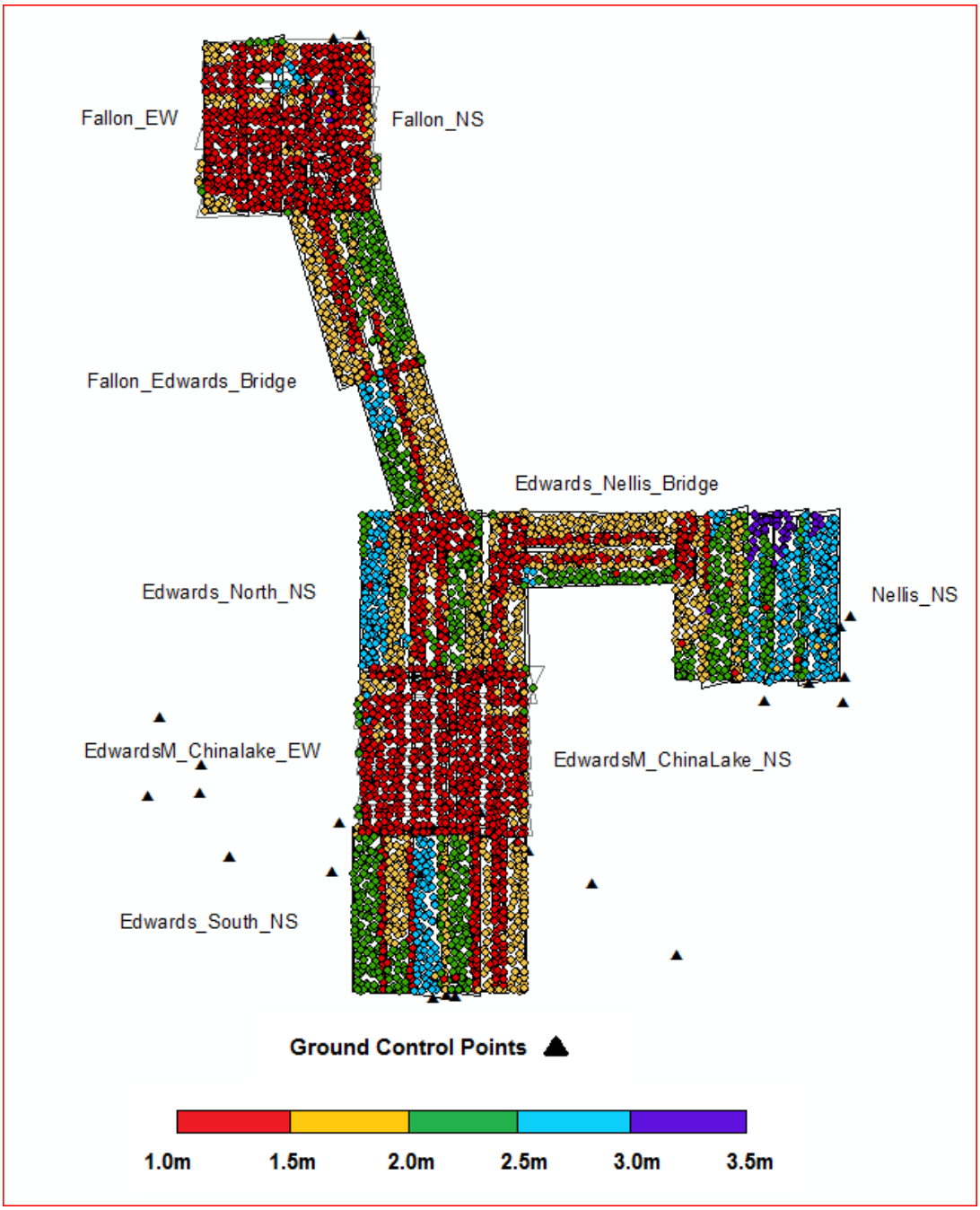

Figure 6: Predicted CE90 per MIN point, in meters.

ATP and MIN processing details

The DigitalGlobe HPC MIN implementation is designed to build MINs on a continental scale; the size of the MIN we can build increases indefinitely with the number of nodes and cores we can throw at the problem. There is no theoretical upper bound to the number of points the DigitalGlobe HPC system can handle; however, in practice, on the storage system we use (a 549 TB Isilon storage system over NFS, which is not high-performance storage), we find ourselves bound by file I/O as our MINs get larger, which reduces our ability to take advantage of multithreading and distributed processing. However, MINs such as this one, on the scale of 4000-5000 points, are small for the current system.

All of the ATP processing for the Edwards AFB study was done on two small clusters of Dell R815 (6 nodes) and Dell 720 (8 nodes) servers. Each Dell R815 server has 4 AMD Opteron 6174 2.2 Ghz 12-core processors (48 cores total) with 256 GB RAM, while each Dell 720 has 4 Intel E5-2690 2.9 Ghz 8-core processors (32 cores total) also with 256 GB RAM. The distributed and parallel processing in ATP is implemented using MPI.

For this run, 19,316 quadkeys were processed containing between 1 and 6 overlapping stereo pairs. The total number of interest points generated for these quadkeys in initial processing is over 732 million. These are filtered and combined to form 860,000 tie point sets, which are ranked and filtered down to 17,000 tie points. These tie points are subsampled to form the final set of 4686 points that are used in MIN update processing. The total (wall clock) time required for ATP processing was roughly 28 hours.

Stages 1 and 2 processing were done on the Dell R815 cluster. All of the processing for Stages 1 and 2 occurred on a

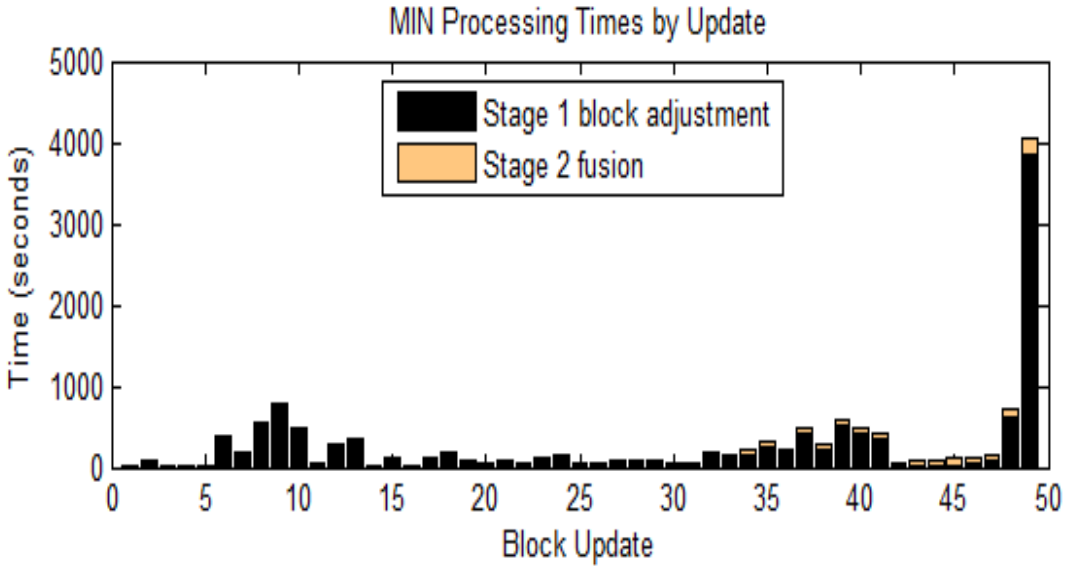

Figure 7: MIN processing times by update. The final update (49) is a fully dense, large bundle adjustment problem.

single core of one server. While our Stage 1 and 2 code is multi-threaded, it is more efficient to run the small problems from earlier updates on a single core because there is slight overhead associated with multithreaded execution over so many cores, which can produce slower overall times initially. We have not yet experimented with optimizing multi-threaded performance over multiple cores, but that is expected to significantly decrease the time associated with massive MINs. Additional optimizations with file I/O are expected to further improve performance.

The 49 updates of Stages 1 and 2 for Edwards AFB took a total of 3.72 hours, which averages 4.5 minutes per update. When excluding the final update, which combined existing MIN points from across the region for a fully dense block adjustment and took over an hour, the average time per update was 3.2 minutes. By comparison, results published in [DS10A-B] for the same area reported about one minute per update on a different computer. Our updates have approximately 3.5 times as many points per update, which would increase the time proportionally more than that.

Wall clock time for both stages is superlinear in the number of points (it is difficult to say precisely, but algebra and storage of key matrices is quadratic in the number of points). One would expect that if we doubled the number of points in the updates, it would take more than twice as long to finish. Indeed, if we were to run with the same number of points as in the experiment in [DS10A-B], in which there are 3.5 times fewer points, our last stage 2 update would take roughly a twelfth of the time.

shows the wall clock time for each update, broken down for Stages 1 and 2. One can see that as the updates progress and the number of MIN points grows, the time for stage 2 increases. However, for this particular MIN, even in the later updates, the bulk of the time is spent in the bundle adjustment. It would take many more updates for Stage 2 to surpass the time of the bundle adjustment. That is the point at which our distributed parallel, multi-threaded HPC pipeline is expected to deliver superior performance.

5. REFERENCES

[CSM07] "Community Sensor Model (CSM) Technical Requirements Document (TRD)", Version 2.A Revision, 1 August 2007.

[DLST06] Dolloff, J., B. Lofy, A. Sussman, and C. Taylor, "Strictly Positive Definite Correlation Functions", *Proc. SPIE* 6235, Signal Processing, Sensor Fusion, and Target Recognition XV, 62351A, May 17, 2006.

[DI07] Dolloff, J., and M. Iiyama (2007), "Fusion of Image Block Adjustments for the Generation of a Ground Control Network," Proceedings from the Information Fusion, 2007 10th International Conference, July 9-12, 2007.

[DIS08] Dolloff, J., M. Iiyama, and R. Settergren (2008), "A New and Efficient Method for the Generation of a Ground Control Network with a Large Area of Coverage", Proceedings from the ASPRS Annual Conference, Portland OR, April 28 – May 2, 2008.

[DS10A] Dolloff, J., and R. Settergren (2010), "WorldView-1 Stereo Extraction Accuracy with and without MIN Processing", Proceedings from the ASPRS Annual Conference, San Diego CA, April 26-April 30, 2010.

[DS10B] Dolloff, J., and R. Settergren (2010), "An Assessment of WorldView-1 Positional Accuracy based on 50 Contiguous Pairs of Stereo Imagery", J. of Photogrammetric Engineering & Remote Sensing (PE&RS), Vol. 76, No. 8, August 2010.

[DG07] DigitalGlobe, *WorldView-1*. Retrieved from http://www.digitalglobe.com/downloads/WorldView1-DS-WV1-Web.pdf.

[DG09] DigitalGlobe, *WorldView-2*. Retrieved from http://www.digitalglobe.com/downloads/WorldView2-DS-WV2-Web.pdf.

[D10] Dolloff, J., "Parameter ranges for a Valid CSM Decay Model", NGA White paper, Case #11-173, July 2010.

[DTL12] Dolloff, J., H. Theiss, and S. Lee, "Generation and Application of RPC Uncertainty Parameters", Appendix F. NGA Technical Document, Case #11-463, October 2012.

[LA09] Lourakis, M., and A. Argyros, "SBA: A Software Package for Generic Sparse Bundle Adjustment", ACM Trans. On Mathematical Software, 2009. 1-20.

[MA82] Mikhail, E., and F. Ackermann, *Observations and Least Squares.* University Press of America, 1982.