JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 113, B12410, doi:10.1029/2008JB005736, 2008

Fixed point theorems of GPS carrier phase ambiguity resolution and their application to massive network processing: Ambizap

Geoffrey Blewitt[1]

Received 5 April 2008; revised 22 July 2008; accepted 3 September 2008; published 23 December 2008.

[1] Precise point positioning (PPP) has become popular for Global Positioning System (GPS) geodetic network analysis because for *n* stations, PPP has $O(n)$ processing time, yet solutions closely approximate those of $O(n^3)$ full network analysis. Subsequent carrier phase ambiguity resolution (AR) further improves PPP precision and accuracy; however, full-network bootstrapping AR algorithms are $O(n^4)$, limiting single network solutions to $n < 100$. In this contribution, fixed point theorems of AR are derived and then used to develop "Ambizap," an $O(n)$ algorithm designed to give results that closely approximate full network AR. Ambizap has been tested to $n \approx 2800$ and proves to be $O(n)$ in this range, adding only ~50% to PPP processing time. Tests show that a 98-station network is resolved on a 3-GHz CPU in 7 min, versus 22 h using $O(n^4)$ AR methods. Ambizap features a novel network adjustment filter, producing solutions that precisely match $O(n^4)$ full network analysis. The resulting coordinates agree to $\ll$1 mm with current AR methods, much smaller than the ~3-mm RMS precision of PPP alone. A 2000-station global network can be ambiguity resolved in ~2.5 h. Together with PPP, Ambizap enables rapid, multiple reanalysis of large networks (e.g., ~1000-station EarthScope Plate Boundary Observatory) and facilitates the addition of extra stations to an existing network solution without need to reprocess all data. To meet future needs, PPP plus Ambizap is designed to handle ~10,000 stations per day on a 3-GHz dual-CPU desktop PC.

Citation: Blewitt, G. (2008), Fixed point theorems of GPS carrier phase ambiguity resolution and their application to massive network processing: Ambizap, *J. Geophys. Res.*, *113*, B12410, doi:10.1029/2008JB005736.

1. Introduction

[2] Since 1994, when the International GNSS Service (IGS) became operational [*Beutler et al.*, 1994; *Dow et al.*, 2005], the analysis of the global GPS network (GGN) by several IGS analysis centers has consistently delivered high-accuracy satellite orbit positions and satellite clock biases. These, in turn, have allowed investigators to compute accurate ground station positions for both regional- and global-scale networks [*Moore*, 2007]. Use of these products have enabled scientific discoveries and monitoring capabilities, with scientific contributions to plate tectonics, the earthquake cycle, glacial isostatic adjustment, crustal and mantle rheology, and surface mass redistribution [e.g., *Blewitt*, 2007].

[3] As of 2008, data from ~2800 continuously operating GPS stations around the world including 400 IGS stations are routinely downloaded from IGS and regional data centers for subsequent analysis at University of Nevada, Reno (UNR) (Figure 1). As full network least squares computations scale as $O(n^3)$, this poses a significant barrier to the full exploitation of all available data. Since its invention by *Zumberge et al.* [1997], PPP has become popular for regional GPS network processing, because processing time scales linearly with the number of stations, $O(n)$, and PPP closely reproduces an $O(n^3)$ full network solution (in fact, it exactly reproduces the solution for the subset of stations used initially for orbit and clock determination).

[4] In GPS positioning, resolution of the integer cycle ambiguity in the carrier phase data can significantly improve positioning precision and accuracy, particularly in the east component for equatorial to midlatitude stations [*Blewitt*, 1989]. Theoretical properties of ambiguity resolution are here exploited to derive a very rapid algorithm, which is then applied to GPS network solutions that have first been derived by precise point positioning (PPP). However, the processing time for full network ambiguity resolution generally scales as $O(n^4)$, thus the main practical advantage of PPP can be lost.

[5] Motivating this study was the idea that theoretical properties of ambiguity resolution might point the way to $O(n)$ processing schemes. A reasonable condition for such schemes to be acceptable is that the differences between optimal and suboptimal solutions should be statistically insignificant ("near optimal"). Here a new algorithm is developed to apply ambiguity resolution to a GPS network with $O(n)$ computation time, which has been demonstrated

[1] Nevada Bureau of Mines and Geology and Seismological Laboratory, University of Nevada, Reno, Nevada, USA.

0148-0227/08/2008JB005736$09.00

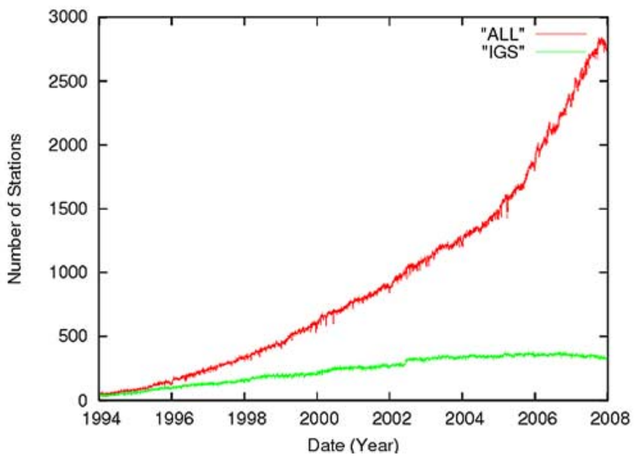

Figure 1. Number of continuous GPS stations per day routinely analyzed at UNR versus date. The upper red curve represents all stations, and the lower green curve the subset that are official IGS stations.

up to n ≈ 3000 at a rate of ~5 s per station on a 3-GHz processor.

2. Theoretical Considerations

2.1. Overview

[6] As a general strategy let us seek to partition an n-station network solution into a number $O(n)$ of self-contained computational blocks giving results that can be assembled as $O(n)$. Our practical definition of $O(n)$ can be relaxed to allow for higher-order effects common for minor necessary tasks, such as $O(n \log n)$ sorting, provided they add a negligible percentage of processing time for $n < 10^4$. As a general guide to developing an accurate algorithm, it is important to consider theoretical properties of GPS networks subject to full network ambiguity resolution. This section briefly discusses theoretical considerations only, so as to clearly separate it from the specific implementation. Although theoretical considerations generally require some level of necessary rigor, the real proof of their validity in this context will be empirical, with demonstrated accuracy and computation times.

[7] In mathematics, a "fixed point theorem" is a statement that, under certain conditions, operator $F(x)$ will have at least one fixed point satisfying $F(x) = x$. A fixed point theorem might also specify what the fixed points are or how to find them [*Shashkin*, 1991]. In the context of this paper, F represents the "bias-fixing operator," which uses ambiguity resolution to adjust double difference biases to perfect values (with zero variance), and thus update other correlated parameters. Here we seek a fixed point for the mapping of parameters from their initial PPP estimates to their bias-fixed estimates.

[8] Since parameter sets can always be transformed into another equivalent set (that is complete and linearly independent), let us consider linear transformations $\tilde{\mathbf{s}} = \mathbf{\Lambda s}$ that satisfy $F(\tilde{\mathbf{s}}) = \tilde{\mathbf{s}}$, for arbitrary values of parameters **s.** As shown in section 2.2, there exists such a fixed point which can be interpreted as the weighted mean centroid of the network. Moreover, it is shown that under conditions common for permanent GPS networks, baselines that have already been bias fixed are insensitive to the bias fixing of other baselines in the network. Taken together, these fixed points suggest a strategy of constructing a network solution out of $n - 1$ bias-fixed baseline vectors (relative coordinates between station pairs), where the initial PPP solutions provide an absolute position to the network.

[9] This section also explores the stochastic nature of large bias-fixed networks, considering that an $O(n)$ algorithm must abandon the computation of the full network covariance matrix. As will be shown, it is remarkable that a block diagonal representation of the covariance matrix is almost exact for large networks. Finally, the selection an optimal set of $n - 1$ baselines is addressed by considering the theory of Euclidean minimum spanning trees, with the goal of selecting an algorithm that contributes a negligible fraction of the overall processing time.

2.2. Fixed Point Theorem 1: Centroid

[10] For an n-station network, let us start with n independent station solutions from PPP, which can be written as vectors $\mathbf{s}_i$ with covariance matrices $\mathbf{C}_i$ for stations $i \in \{1,\ldots,n\}$. Let vectors $\mathbf{s}_i$ include station coordinates and single-difference carrier phase biases between all satellites in common view. The dimensions of all $\mathbf{s}_i$ and the order of parameters in $\mathbf{s}_i$ are assumed to be identical for all stations. (The consequences of noncommon visibility will be addressed in paragraph 16.)

[11] Let us define the bias fixing operator $F(\mathbf{\Lambda s})$ as the mapping of any linear combination of parameters from their initial PPP solution to a bias-fixed solution, as a result of ambiguity resolution of differences in the single-difference biases. Now the first fixed point theorem is stated:

$$\text{Theorem 1} \qquad \bar{\mathbf{s}} \equiv \bar{\mathbf{C}} \sum_i \mathbf{C}_i^{-1} \mathbf{s}_i \Rightarrow F(\bar{\mathbf{s}}) = \bar{\mathbf{s}} \tag{1}$$

where $\bar{\mathbf{C}} \equiv \left[\sum_i \mathbf{C}_i^{-1}\right]^{-1} = \mathrm{Var}(\bar{\mathbf{s}})$. By definition, the covariance matrices $\mathbf{C}_i = \mathrm{Var}(\mathbf{s}_i)$ are understood to be constant, referring to the values given by PPP (but the values of $\mathbf{s}_i$ in equation (1) can be from either before or after bias fixing). Simply put, the weighted mean parameter vector ("centroid") is a fixed point with respect to bias fixing.

[12] It is sufficient to prove that the centroid is not correlated with differences in station parameters, by invoking the block diagonal nature of the formal covariance matrix from PPP:

$$\begin{aligned}
\mathrm{Cov}(\bar{\mathbf{s}}, \mathbf{s}_i - \mathbf{s}_j) &= \bar{\mathbf{C}} \sum_k \mathbf{C}_k^{-1} \left[\mathrm{Cov}(\mathbf{s}_k, \mathbf{s}_i) - \mathrm{Cov}(\mathbf{s}_k, \mathbf{s}_j)\right] \\
&= \bar{\mathbf{C}} \sum_k \left[\mathbf{C}_k^{-1} \delta_{ki} \mathbf{C}_i - \mathbf{C}_k^{-1} \delta_{kj} \mathbf{C}_j\right] \\
&= \bar{\mathbf{C}} \left[\mathbf{C}_i^{-1} \mathbf{C}_i - \mathbf{C}_j^{-1} \mathbf{C}_j\right] \\
&= 0
\end{aligned} \tag{2}$$

hence proving theorem 1.

[13] Now let us assume the lemma that correlations between two variables will remain zero if a new measurement is a function of only one of the two variables. Since

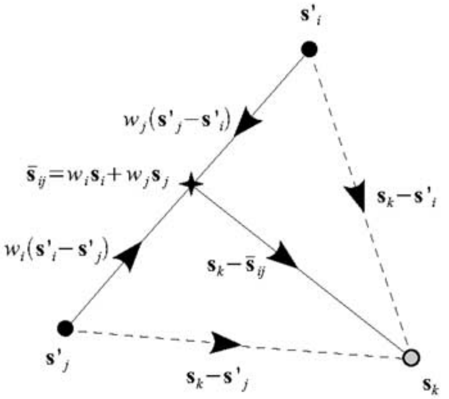
$\mathbf{s}'_i$
$w_j(\mathbf{s}'_j - \mathbf{s}'_i)$
$\bar{\mathbf{s}}_{ij} = w_i\mathbf{s}_i + w_j\mathbf{s}_j$
$\mathbf{s}_k - \mathbf{s}'_i$
$w_i(\mathbf{s}'_i - \mathbf{s}'_j)$
$\mathbf{s}_k - \bar{\mathbf{s}}_{ij}$
$\mathbf{s}'_j$
$\mathbf{s}_k - \mathbf{s}'_j$
$\mathbf{s}_k$

Figure 2. Diagram illustrating the geometry used in the proof of theorem 2, showing the bias fixing of a third station $\mathbf{s}_k$ to an existing bias fixed baseline ($\mathbf{s}'_i - \mathbf{s}'_j$), by resolving ambiguities to the centroid of that baseline $\bar{\mathbf{s}}_{ij}$.

ambiguity resolution represents only a measurement of parameter differences, this leads to the corollary:

$$\mathrm{Cov}\left(\bar{\mathbf{s}}, \mathbf{s}'_i - \mathbf{s}'_j\right) = 0 \qquad (3)$$

where the primes indicate solutions after ambiguity resolution.

[14] Another corollary of the theorem is that the variance of the centroid $\bar{\mathbf{C}} = \mathrm{Var}(\bar{\mathbf{s}})$ is not changed by F. Therefore, to compute the absolute location of a bias fixed network, all that is required is already available in the form of n statistically independent PPP solutions and $O(n)$ computations.

[15] One problem in applying equation (1) is that the station coordinate components of the fixed point vector are linear combinations of all parameters, including station coordinates and bias parameters. It would be much more convenient if a weighted average of station coordinate triplets alone provided a fixed point. It is now shown that this condition is satisfied when the PPP covariance matrices for all stations are the same to within any positive scale factor. For this "assumption of similar covariances," let us write

$$\mathbf{C}_i^{-1} = w_i\bar{\mathbf{C}}^{-1} \qquad (4)$$

where the scalar weights satisfy $\sum_i w_i = 1$. In this case, equation (1) reduces to

$$\bar{\mathbf{s}} \equiv \sum\nolimits_i w_i\mathbf{s}_i \qquad (5)$$

As the order of parameters are the same in all vectors $\mathbf{s}_i$ and $\bar{\mathbf{s}}$, the computations of weighted average of individual parameters are decoupled in equation (5). Therefore, under the assumption of similar covariances, the weighted mean station coordinates can be computed without reference to the bias parameters.

[16] The assumption of similar covariances requires that observation schedules for bias-fixed baselines are similar (but the data rates do not need to be the same), and that there is reasonable common visibility of satellites. In practice it is recommended that algorithms derived from this theory only be applied to sites with data sets of equal duration, for example, continuously operating sites with full (or nearly full) 24-h data sets. It is also recommended that algorithms be designed to select nearest neighbor stations to conduct ambiguity resolution, both to maximize common visibility, and to maximize probability of success in ambiguity resolution.

[17] Since the absolute position of a bias fixed network can be computed as $O(n)$, what remains is the computation of the relative positions of stations in the network, which theoretically is completely specified by $n - 1$ baseline vectors. This suggests that $O(n)$ computation of the network may be possible given the computation of $n - 1$ suitable bias-fixed baseline vectors. "Suitable" baseline vectors would need to replicate, or very closely approximate, the relative coordinates derived by full network ambiguity resolution. Furthermore, assuming such suitable baseline vectors can be computed, theorem 1 implies that to compute $\bar{\mathbf{s}}$, as defined by equation (1), we can choose to use either the original PPP solutions $\mathbf{s}_i$ or the solutions $\mathbf{s}' = F(\mathbf{s})$ derived from the suitable baseline vectors, together with the original PPP covariance matrices.

2.3. Fixed Point Theorem 2: Baselines

[18] Consider a network solution initialized by PPP, where a single baseline is then bias fixed to produce parameter vectors $\mathbf{s}'_i$ and $\mathbf{s}'_j$, which include both station coordinates and biases. As explained previously, let us make the assumption of similar covariances. Let the bias fixing operator F (defined in section 2.1) then be applied to the entire network.

$$\text{Theorem 2} \qquad \hat{\mathbf{s}}'_{ij} \equiv \mathbf{s}'_i - \mathbf{s}'_j \Rightarrow F\left(\hat{\mathbf{s}}'_{ij}\right) = \hat{\mathbf{s}}'_{ij} \qquad (6)$$

Simply put, the bias-fixed solution of baseline parameters is independent of bias fixing of other baselines in the network.

[19] To prove this, consider bias fixing stations i and j to an independent PPP solution $\mathbf{s}_k$ for any $k \notin \{i, j\}$, as shown in Figure 2. From equation (3),

$$\mathrm{Cov}\left(\mathbf{s}_k - \bar{\mathbf{s}}_{ij}, \mathbf{s}'_i - \mathbf{s}'_j\right) = 0 \qquad (7)$$

where $\bar{\mathbf{s}}_{ij}$ is the centroid of stations i and j, defined by equation (1). Thus, adding any independent information on ($\mathbf{s}_k - \bar{\mathbf{s}}_{ij}$) (alone) has no effect on $\hat{\mathbf{s}}'_{ij}$. Expanding the term ($\mathbf{s}_k - \bar{\mathbf{s}}_{ij}$) results in the weighted average of two baseline parameter vectors from station k:

$$\begin{aligned}\mathbf{s}_k - \bar{\mathbf{s}}_{ij} &= \mathbf{s}_k - \bar{\mathbf{C}}\sum_{p\in\{i,j\}}\mathbf{C}_p^{-1}\mathbf{s}_p \\ &= \bar{\mathbf{C}}\mathbf{C}_i^{-1}\left(\mathbf{s}_k - \mathbf{s}'_i\right) + \bar{\mathbf{C}}\mathbf{C}_j^{-1}\left(\mathbf{s}_k - \mathbf{s}'_j\right)\end{aligned} \qquad (8)$$

Now applying equation (4) (the assumption of similar covariances), equation (8) becomes

$$\mathbf{s}_k - \bar{\mathbf{s}}_{ij} = w_i\left(\mathbf{s}_k - \mathbf{s}'_i\right) + w_j\left(\mathbf{s}_k - \mathbf{s}'_j\right) \tag{9}$$

where $w_i + w_j = 1$ and $\bar{\mathbf{s}}_{ij} = w_i\mathbf{s}_i + w_j\mathbf{s}_j$. As was the case for section 2.2, the assumption of similar covariances decouples the linear combinations of biases from the station coordinates, so making it possible to resolve ambiguities and bias fix equation (9). There are only two linearly independent baselines in a three-station subnetwork, and one of those baselines $(\mathbf{s}'_i - \mathbf{s}'_j)$ is already bias fixed; thus the other two baselines can be constructed as linear combinations of $(\mathbf{s}'_i - \mathbf{s}'_j)$ and $(\mathbf{s}_k - \bar{\mathbf{s}}_{ij})$:

$$\begin{aligned}\left(\mathbf{s}_k - \mathbf{s}'_i\right) &= \left(\mathbf{s}_k - \bar{\mathbf{s}}_{ij}\right) + w_j\left(\mathbf{s}'_j - \mathbf{s}'_i\right)\\ \left(\mathbf{s}_k - \mathbf{s}'_j\right) &= \left(\mathbf{s}_k - \bar{\mathbf{s}}_{ij}\right) + w_i\left(\mathbf{s}'_i - \mathbf{s}'_j\right)\end{aligned} \tag{10}$$

Therefore additional bias fixing of $(\mathbf{s}'_k - \bar{\mathbf{s}}'_{ij})$ is equivalent to bias fixing the entire three-station subnetwork (Figure 2). This fact, together with equation (7) demonstrates that the first bias fixed baseline vector is insensitive to additional bias fixing in the network.

[20] Note that equation (9) was only constructed to prove the theorem, so we do not literally need to bias fix this linear combination of baselines. According to the theorem, baselines can be individually bias fixed, then combined together to approximate closely the full network solution, provided the conditions of common visibility outlined in section 2.2 are closely met.

2.4. Stochastic Properties of Large Bias-Fixed Networks

[21] Consider the cross covariance between station positions of different stations. To get an idea of how this cross covariance changes a function of the number of bias-fixed stations n, let us assume that all station covariance matrices from PPP are identical $\mathbf{C}_i \equiv \mathbf{C}$, which allow us to write the inverse variance of the centroid as a function of n:

$$\bar{\mathbf{C}}^{-1}(n) = n\mathbf{C}^{-1} \tag{11}$$

Now let us assume that, after bias fixing, all cross covariances are equal, and all variances are equal, and assume they are functions of n:

$$\begin{aligned}\mathbf{X}(n) &\equiv \mathrm{Cov}\left(\mathbf{x}'_i, \mathbf{x}'_{j\neq \mathrm{i}}\right)\\ \mathbf{V}(n) &\equiv \mathrm{Var}\left(\mathbf{x}'_i\right) = \mathrm{Cov}\left(\mathbf{x}'_i, \mathbf{x}'_i\right)\end{aligned} \tag{12}$$

which uses the previous result that the variance matrix of the centroid remains constant under bias fixing. As a corollary of theorem 2, the variance $\mathbf{B}$ of the relative position for a baseline that is already bias fixed remains constant as the network grows:

$$\mathbf{B} \equiv \mathrm{var}\left(\mathbf{x}'_i - \mathbf{x}'_{j\neq \mathrm{i}}\right) = 2[\mathbf{V}(n) - \mathbf{X}(n)] \tag{13}$$

The following lemma will now be used, which can be verified by taking the mean of both sides (which is allowed because the left hand side must be identical for all stations) and by applying $\bar{\mathbf{x}}' = \bar{\mathbf{x}}$ (from theorem 1):

$$\mathrm{Cov}\left(\mathbf{x}'_i, \bar{\mathbf{x}}\right) = \mathrm{Var}(\bar{\mathbf{x}}) = \bar{\mathbf{C}} \tag{14}$$

Expanding both sides by substituting equations (11) and (12) gives us

$$\begin{aligned}\bar{\mathbf{C}} &= (1/\mathrm{n})\mathbf{C}\\ &= \mathrm{Cov}\left(\mathbf{x}'_i, \bar{\mathbf{x}}\right)\\ &= (1/\mathrm{n})\sum_j \mathrm{Cov}\left(\mathbf{x}'_i, \mathbf{x}'_j\right)\\ &= (1/\mathrm{n})[\mathbf{V}(n) + (n-1)\mathbf{X}(n)]\end{aligned} \tag{15}$$

Substituting equation (13) and rearranging gives

$$\begin{aligned}\mathbf{X}(n) &= \frac{1}{n}(\mathbf{C} - \mathbf{B}/2)\\ \lim_{\mathrm{n}\to\infty} \mathbf{X}(n) &= 0\end{aligned} \tag{16}$$

$$\begin{aligned}\mathbf{V}(n) &= \mathbf{B}/2 + \frac{1}{n}(\mathbf{C} - \mathbf{B}/2)\\ \lim_{n\to\infty} \mathbf{V}(n) &= \mathbf{B}/2\end{aligned} \tag{17}$$

where the constant term $(\mathbf{C} - \mathbf{B}/2) > 0$ is equal to half the variance reduction in relative position due to bias fixing, and so is positive definite. Thus $\mathbf{X}(\infty) = 0$ and $\mathbf{V}(\infty) = \mathbf{B}/2$, which is half the variance of the bias-fixed relative position.

[22] Thus the correlation between different station positions is inversely proportional to the number of stations, and becomes negligible for large networks $n > 10^2$ (as can be verified in practice). Therefore, in the limit of large n, the ambiguity resolution from a new station to a large network only affects that station's coordinates.

[23] The above stochastic properties of large bias-fixed networks lend further evidence to suggest that accurate $O(n)$ algorithms are feasible. Clearly, $O(n^4)$ full network algorithms when applied to large networks waste their time computing the off-diagonal elements of the covariance matrix, when theory predicts that they tend to vanish and so hold negligible information content. This suggests it is reasonable to represent the final covariance matrix of station coordinates as block diagonal (of triplets), for which the number of computed elements is $O(n)$ (as is the case for PPP).

2.5. Theory of Optimal Baseline Selection

[24] As the goal is to maximize the probability of correctly resolving the ambiguities, let us select the set of $n - 1$ baselines that minimize the sum of distances between n stations (a geometrical version of the "traveling salesman problem"). This is known as the Euclidean minimum spanning tree (EMST). An exact solution to the EMST is given by Kruskal's "greedy" algorithm [*Kruskal*, 1956], which starts with each station as its own disjoint tree in a "forest" (the union of all trees), then grows the trees by iteratively adding the next shortest baseline that does not destroy the tree by forming a cycle (i.e., does not already have both stations within the same tree). The algorithm stops when all stations are in one tree. The problem is that

Kruskal's algorithm is unacceptably slow at $O(b\log n) = O(n^2\log n)$ (R. Sedgewick and K. Wayne, Minimum spanning tree, lecture notes for Computer Science Course 226: Algorithms and Data Structures, 2007, available at http://www.cs.princeton.edu/courses/archive/fall07/cos226/lectures.html), where $b = n(n - 1)/2$ is the number of possible baselines that can be formed from n stations.

[25] The solution to this problem uses the Delauney triangulation on the sphere, which can be computed in $O(n\log n)$ [*Renka*, 1997]. The Delauney triangulation is the mathematical dual of the Vornoi diagram, which is constructed of n polygons, each centered on station, defining the nearest neighbor station for every possible query point [*Aurenhammer*, 1991]. Station pairs are defined as nearest neighbors if their geodesic crosses only one shared Voroi edge. This defines a $b \leq (3n - 6)$ set of baselines [*Renka*, 1997].

[26] The Delauney triangulation has the relevant mathematical property that is a supergraph of the EMST [*Aurenhammer*, 1991]. This means that all baselines of the EMST are baselines of the Delauney triangulation of the stations. Therefore the EMST solution of $n - 1$ baselines is a subset of the $b \leq (3n - 6)$ baselines from the Delauney triangulation, so the problem reduces to finding this subset. It follows that Kruskal's algorithm can be used to find the EMST as a subset of the Delauney triangulation in $O(b\log n) = O(n\log n)$. Therefore the combined algorithm is also $O(n\log n)$ (where $\log n < 4$ in our GPS universe). This not quite the $O(n)$ theoretical performance we seek, however in practice it proves to add negligible ($\ll$1%) computation time for networks of $n < 10^4$. This is because the computation time is completely dominated by the bias fixing of $n - 1$ independent baselines, even though this part of the computation is $O(n)$.

[27] For our problem, optimal baseline selection is a little more complicated than solving for the EMST, because it might not be possible to resolve ambiguities successfully on a specific baseline, and alternative spanning trees must be found. When ambiguity resolution fails, a pitfall to avoid is the testing all possible alternative baselines, as this could end up being an $O(n^2)$ computation. Fortunately, the Delauney triangulation limits the number of baselines to $b \leq (3n - 6)$, and so keeps the computation at $O(n\log n)$.

[28] Interestingly, this points to a mechanism that might result in overall computation time better than $O(n)$ (which is seemingly impossible). Consider the case of globally distributed stations. As the number of stations n increases, so the average baseline length decreases, and so the number of ambiguity resolution failures decrease. Therefore, the number of baselines that require bias fixing computations might range from the Delauney triangulation limit of $b \approx (3n - 6)$ for small n, to $b \approx n - 1$ for large n. Since bias fixing dominates the computation time, in theory it is possible to have computation times smaller than $O(n)$ (as a general rule, because in practice, this will depend on specific details of the network geometry).

2.6. Summary of Theoretical Results

[29] The following now summarizes what has been learned from theoretical considerations, under assumptions that should be reasonably well satisfied by continuous GPS networks. (1) When bias fixing a network (or partly bias fixing anywhere inside a network), the centroid of that network remains fixed. (2) Estimates of the bias-fixed relative coordinates between any pair of stations are insensitive to bias fixing elsewhere in the network. (3) As n becomes large, the final covariance matrix resulting from full network bias fixing tends toward a block diagonal structure. (4) Selection of an optimal set of $n - 1$ baselines to connect the network can be computed in $O(n\log n)$, which for $n < 10^4$ has a computation time that is negligible compared to $O(n)$ bias-fixing computations.

[30] Synthesizing these theoretical results brings the conclusion that independent bias fixing of $n - 1$ baselines together with initial PPP covariance matrices for each station can be used to construct the full network bias fixed solution for n stations and covariance matrix to a very good approximation. This summarizes the theoretical rationale for the design of the Ambizap algorithm, which is the topic of section 3.

[31] The stated assumptions led to recommendations as to situations when this theory may or may not be applicable, with the key recommendation being that bias fixing should be applied to the shortest baselines between stations that have the same nominal observation schedules. Ultimately, the validity of applying the theory in practice must be proved empirically, as shown in section 4.

3. Implementation: Ambizap

3.1. Design Overview

[32] As discussed in section 2.6, the fixed point theorems imply that a complete network solution can be constructed as $O(n)$ by a two-step procedure: (1) bias fix the vectors of $n - 1$ linearly independent baselines and (2) perform a network adjustment that minimizes distortion in the bias-fixed baselines, while maintaining alignment with the original PPP solutions. The output covariance matrix from the network adjustment should only include the block matrices for each individual station. This suggests that the network adjustment in step 2 should be performed using blocking techniques to avoid unnecessary computation of off-diagonal covariance elements. Having a minimal set of $n - 1$ baselines, in turn, suggests a kind of network estimation filter that steps through the network tree, baseline-by-baseline (analogous to the more familiar epoch-by-epoch Kalman filter).

[33] The algorithm Ambizap has been encoded (and made freely available to researchers), which is a stand-alone computer program consisting of a C-shell script driving FORTRAN-compiled executables. The software reads in individual station PPP solutions, and outputs individual station bias-fixed solutions (including covariance matrices) that closely approximate the output of an $O(n^4)$ full network ambiguity resolution analysis. The software reads and writes data files in formats consistent with GIPSY OASIS II, but in principle the algorithm could be adapted to work with any PPP-capable software. The only internal dependence on GIPSY OASIS II modules is the core ambiguity resolution engine, "Ambigon" [*Blewitt*, 1989] which is only applied at the single baseline level. In principle, Ambigon could be substituted for another core engine.

[34] The three key design requirements of Ambizap were that (1) except for ancillary tasks that take negligible

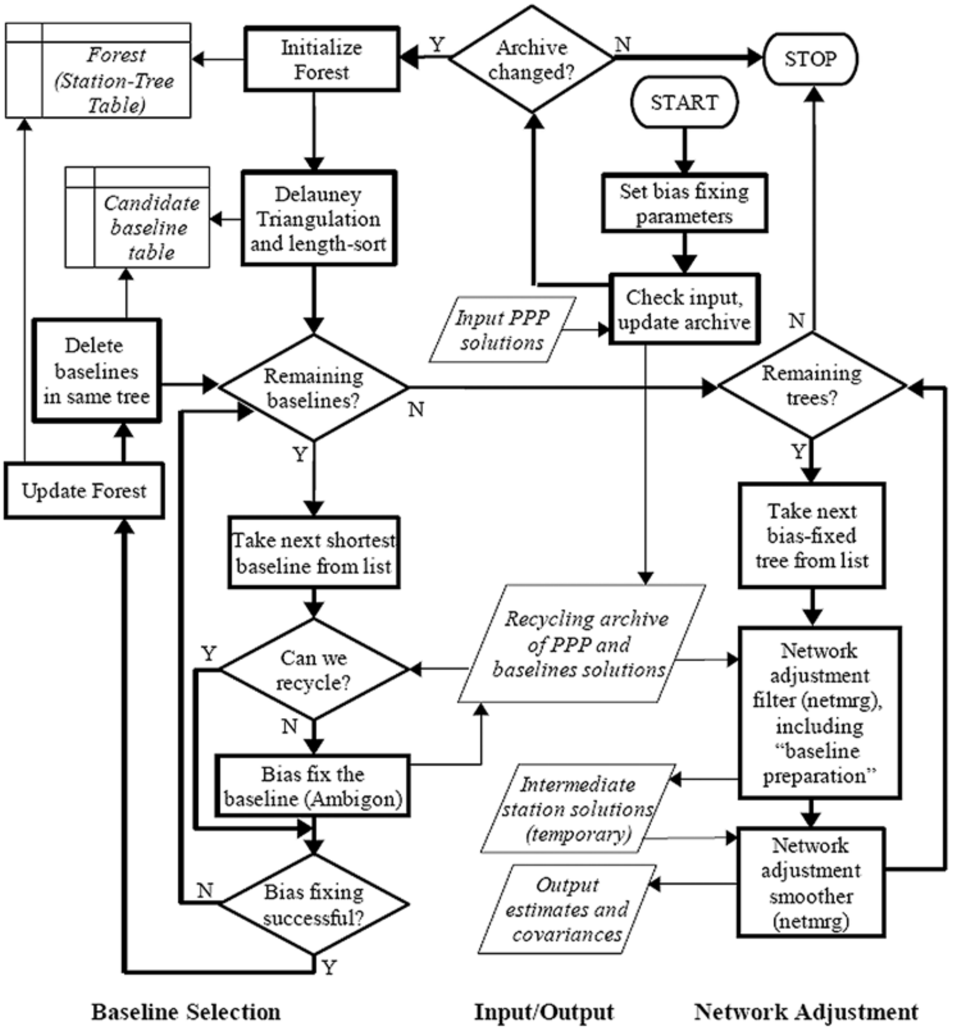
Forest (Station-Tree Table)
Initialize Forest
Y
Archive changed?
N
STOP
START
Candidate baseline table
Delauney Triangulation and length-sort
Set bias fixing parameters
Check input, update archive
Input PPP solutions
Delete baselines in same tree
Remaining baselines?
N
Y
Remaining trees?
N
Y
Update Forest
Take next shortest baseline from list
Take next bias-fixed tree from list
Recycling archive of PPP and baselines solutions
Y
Can we recycle?
N
Network adjustment filter (netmrg), including "baseline preparation"
Bias fix the baseline (Ambigon)
Intermediate station solutions (temporary)
N
Bias fixing successful?
Y
Output estimates and covariances
Network adjustment smoother (netmrg)
Baseline Selection
Input/Output
Network Adjustment

Figure 3. Ambizap flowchart. The start and stop of the flowchart are at the top right.

computation time for $n < 10^4$, overall computation time should be $O(n)$ to accommodate all (a few thousand) permanent GPS geodetic stations in the world today; (2) the resulting station coordinate solutions and formal errors should closely (<1 mm) agree with those produced by full network ambiguity resolution, at least up to maximum number of stations that can be processed this way ($n \sim 100$ previously being a practical limit for GIPSY OASIS II alone); and (3) adding extra stations to or removing unwanted stations from an existing network solution should give an identical answer as computing a new solution from scratch, but should be much faster to compute, for example, by recycling old bias-fixed baseline solutions.

[35] Figure 3 shows an overview of the design of Ambizap. The remainder of this section details the following three key modular components: (1) the selection of $n - 1$ linearly independent baselines; (2) the preparation of the $n - 1$ bias-fixed baseline solutions to facilitate implementation of theorem 2, in which a loosening transformation is applied to each station pair covariance matrix, so that baseline solutions can be combined into a network without affecting the relative coordinates; and (3) network adjustment of $n - 1$ loose baseline solutions, together with a tightening transformation using the n original PPP coordinate covariance matrices in order to compute the final coordinate covariance matrices for each bias-fixed station.

3.2. Baseline Selection

[36] The following steps now describe how baselines are selected, given the possibility of ambiguity resolution failure, based on concepts of section 2.5. These steps are illustrated on the left-hand side of Figure 3.

[37] 1. Initialize algorithm parameters relating to ambiguity resolution. Default values are (1) the cumulative confidence limit at which ambiguity resolution is considered "successful" for a baseline $C_{\min} = 99.5\%$ (otherwise, the bias is left at its real-valued estimate), and (2) the minimum rate of success for a given baseline $S_{\min} = 50\%$, such that if $S < S_{\min}$, then that baseline is dropped for consideration from candidate baselines, and a different path is chosen to connect the network. Here S is defined as the fraction of double difference biases for which $C > C_{\min}$.

[38] 2. Check input files against the contents of the recycle archive, and stop if there is nothing new. Update the archive with any new input PPP data; delete from the archive any PPP data or bias-fixed baselines (from previous

Ambizap runs) that either do not match input station names, or have different input PPP data for either station; then begin processing. This feature recycles old computations, while ensuring consistency with new input data.

[39] 3. Initialize a "disjoint set data structure" [*Galil and Italiano*, 1991] representing the forest of disjoint trees, where each tree is a set of bias-fixed stations. This will guarantee linear independence in the list of up to $n - 1$ selected baselines. This is implemented as a tabular listing of station names and bias-fixed tree number, which are initially all unique $(1,\ldots,n)$, indicating that nothing is yet bias fixed. A "union find" algorithm will be applied whenever a baseline is resolved that connects the two trees i and j (thus bias fixing the union of the two trees). This algorithm works by first finding i and j by matching the names of the stations with the baseline (which, in turn, might be identified by a filename in an archive of previous baseline computations), then forming a union by setting all $j = i$ in the tabublation.

[40] 4. Initialize a list of candidate baselines in the form of two station names and the baseline length. This is achieved by applying the Delauney triangulation on the surface of a sphere, using the STRIPAK subroutine library by *Renka* [1997]. The triangulation has the feature that it does not span any hemisphere that is devoid of stations. The software was modified to allow for multiple stations at the same coordinates ("zero baselines"), which can happen when GPS receivers are attached to the same antenna. With negligible computation time, this selects $b \leq (3n - 6)$ baselines, where $b = (3n - 6) \approx 3n$ if there is no empty hemisphere, which provides redundancy in case of ambiguity resolution failure. This list is then sorted in increasing order of length.

[41] 5. Take the shortest candidate baseline on the list, and go to the next step if a valid recycled solution exists. Otherwise, concatenate the two independent (uncorrelated) PPP station solutions, and then attempt to resolve as many double-differenced ambiguities as possible. The method implemented is the standard ambiguity resolution approach of forming the "wide lane" and the "narrow lane" linear combinations [*Blewitt*, 1989; *Dong and Bock*, 1989]. The wide lane method is automatically selected [*Blewitt*, 1989] from either the ionospheric minimum method or four-observable method [*Melbourne*, 1985; *Wübbena*, 1985]. The narrow lane method is based on ionosphere-free observations, with bootstrapping applied to all satellites observed by both stations, such that resolution of the best determined ambiguities improves the estimates of all remaining ambiguities for that baseline. Note that any method could be used at the core of this modular algorithm, including, for example, the Lambda method [*Teunissen*, 1995]. Resolution of the ~30 to 50 double-differenced biases for a given baseline continues sequentially using bootstrapping until the cumulative confidence $C > C_{\min}$ (multiplied over all double difference biases). In practice for the case of a global network, it is far more likely that any given baseline will have $C > C_{\min}$ for $S = 100\%$ of its biases when $n > 10^3$.

[42] 6. If the percentage of ambiguities resolved $S > S_{\min}$, consider the baseline to be successfully resolved, and perform a union find operation to update the forest (see step 3).

[43] 7. Whether successful or not, eliminate this baseline from the candidate list, and if successful, eliminate all other baselines that have both stations in this tree (see step 3).

[44] 8. Iteratively loop back to (5), etc., until either $n - 1$ baselines have been resolved, or until the candidate list of baselines has been exhausted. Apart from the bias fixing aspects, this iterative loop is Krustal's algorithm.

[45] Note that in the above approach, it is possible to end up having a forest of disconnected trees of bias fixed baselines. Within each tree, biases are effectively resolved between any pair of stations. Therefore network adjustment will be performed independently for each such tree.

3.3. Baseline Preparation

[46] For two reasons, it would be a mistake to perform a traditional weighted least squares network adjustment by combining the $n - 1$ bias-fixed baseline solutions: (1) in violation of theorem 1, such an approach would give multiple weight to PPP solutions from stations associated with multiple baselines (a number that on average is ~2, and rarely exceeds 5 in practice); and (2) in violation of theorem 2, baselines combined in such a manner will in general change their solution significantly as they are combined. Theorems 1 and 2 model the behavior of a full network solution (accounting for all correlations in the network biases and coordinates) and so our algorithm needs to emulate this. Fortunately, the theorems are simple to implement, in that the algorithm simply needs to minimize the distortion of baselines as they are combined, while retaining the mean position of the network.

[47] Implementation of a network adjustment in accordance with theorems 1 and 2 is facilitated by preparation of the input data to the adjustment, which begins in the raw form of the initial PPP estimates and covariance matrices, and the $n - 1$ bias-fixed station pair estimates and covariance matrices. "Measurement downdating" is applied to each covariance matrix $\mathbf{C}'_{i \otimes j}$ for bias-fixed station pairs i and j, which subtracts the weight associated with the original PPP solutions (without changing the estimates themselves).

$$\mathbf{A}_{i\otimes j} = \left[\mathbf{C}'^{-1}_{i\otimes j} - (1-\varepsilon)\begin{pmatrix} \mathbf{C}_i^{-1} & 0 \\ 0 & \mathbf{C}_j^{-1} \end{pmatrix}\right]^{-1} \tag{18}$$

The resulting station pair covariance matrix $\mathbf{A}_{i \otimes j}$ represents the pure information content that bias fixing adds to the original PPP solution for the station pair.

[48] Setting the parameter $\varepsilon = 10^{-4}$ ensures numerical stability given that there is a rank 1 deficiency predicted by theorem 1 (that bias fixing adds no information on the centroid). This (near) rank deficiency is desirable in this case, as it produces a solution that is relatively free to translate without distorting the internal geometry (relative coordinates). Given this preparation, a conventional weighted least squares combination of such loose baseline solutions would produce a unique network solution that remains centered on the original estimates (theorem 1), while retaining the internal geometry of its bias-fixed baselines (theorem 2).

[49] The final covariance matrix for the entire network can in principle be constructed by adding back the PPP

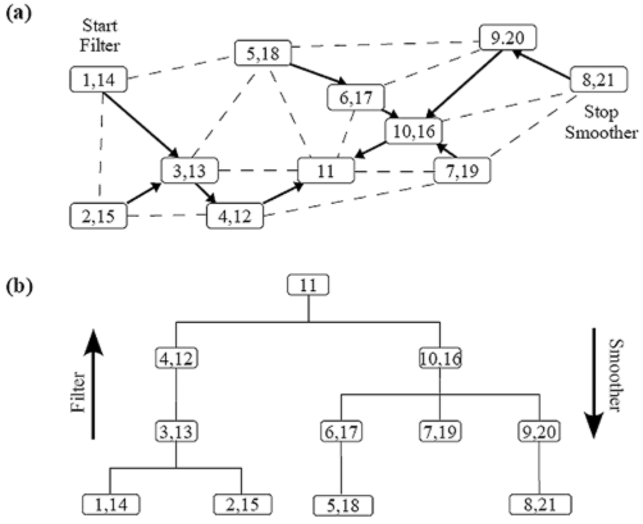

Figure 4. Diagram illustrating the network adjustment filter/smoother structure: (a) physical network showing the Delauney triangulation, where selected baselines (here, the EMST) are arrows that indicate the flow of information in the filter (opposite for the smoother); (b) logical network as a tree. The numbers indicate the order of processing, the first number being for the filter, the second for the smoother. The root station "11" (at the top of the tree) has the last filter solution, which initializes the first smoother solution.

weights either after the estimates have been obtained, or in such a manner that the estimates are not themselves affected by this procedure. As explained in section 3.4, the latter method has been developed because it only requires the computation of covariance matrices for each station.

[50] In summary, the outputs of this step of the algorithm are (1) the input bias-fixed station pair solutions (unchanged), (2) each with a covariance matrix that has been downdated using the PPP covariance matrix, and (3) the original PPP covariance matrices. These data are then used as input to the network adjustment algorithm. In Ambizap (version 2.0), this preparation step is actually integrated into the network adjustment equations to reduce computations. However, it is useful here to have separated this step conceptually, considering it is the part of Ambizap that uses the fixed point theorems, and so makes it fundamentally different than conventional least squares combination.

3.4. Network Adjustment

[51] Taking the inputs described in section 3.3, the goal of network adjustment is to output final bias-fixed solutions for individual stations, including estimates and covariance matrices that closely approximate that of a full network solution. The previous preparation enables conventional least squares to be applied, with the exception that there needs to be an additional customized step to compute the final covariance matrix for each station. As indicated in Figure 3, network adjustment is applied independently to each successfully bias-fixed tree.

[52] A practical problem is that a conventional network adjustment is typically $O(n^3)$ unless the sparse nature of the design matrix is exploited (for example, Helmert blocking). The implemented method takes the blocking concept to its logical extreme, in the form of a network estimation filter that steps through the network tree adding baseline information at each step, and taking care not to count PPP information twice. This method is extreme, in that it only produces a single block matrix for each station, which in our case, is precisely what we are looking for. Even though the full covariance matrix is not computed, the actual estimates are exact (just as Helmert blocking and Kalman filtering are exact), and the computation is $O(n)$ (analogous to Kalman filtering, where n is the number of epochs). It turns out that network adjustment using the following filtering approach takes $\ll 1\%$ of the computation time of $O(n)$ bias fixing for $n - 1$ baselines, and so as a whole takes negligible time. (This stands in contrast to prototype implementations of Ambizap using conventional $O(n^3)$ least squares network adjustment, which begins to dominate the total computation time at $n \approx 10^3$).

[53] The network adjustment algorithm is now summarized here in conceptual terms. The algorithm is a filter/smoother that operates on a tree structure (Figure 4) defined by n stations connected by $n - 1$ baselines. Any of the stations can be arbitrarily selected as the "root" (top) of the tree, which then descends by connected baselines, branching out at junctions, until finally each subbranch is terminated by a singly connected node, each representing a "leaf" of the tree.

[54] Filtering begins at these leaf stations, and adds information as the filter moves up the tree. At each junction, the filter combines information between different branches. When the filter reaches the top of the tree, it has found the final solution for the root station (and only the root station).

[55] Using this solution as a priori information, the filter then goes backward down the tree, an operation called "smoothing" in filtering theory, or "back-substitution" in blocking theory. Information is then added as the smoother moves down the tree, at each step writing out the final solution for each station encountered. When a leaf station is encountered, the smoother jumps back to the last junction and continues until all leaf stations solutions have been computed and written to individual station solution files.

[56] A modification is made to the smoothing algorithm that allows for computation of the final station covariance matrix without disturbing the parameter estimates. This is achieved adding the PPP covariance in the smoother, starting with the root station.

[57] It is the $O(n)$ steps in the structure of the overall computation that makes the filter so fast, and the method used to add information at each step on the tree is not important. Any manner of Kalman filter or square root information filter would accomplish the task without any significant effect on the overall performance of Ambizap. In Ambizap, conventional least squares is applied at each step to simplify readability and maintenance of the code, as it only involves inversions of 6 × 6 matrices, with negligible computation time.

[58] Testing proves that, with the exception of the intentional effect of baseline preparation discussed previously, the solution is identical to that of a conventional $O(n^3)$ least squares combination of baselines. (Of course, only block diagonal elements of the covariance matrix are computed, but this does not limit solution accuracy). The solution is

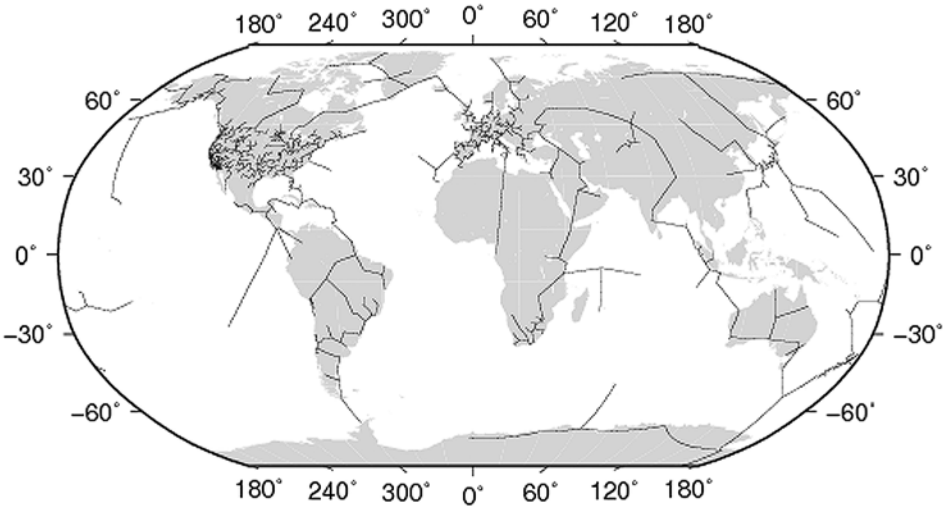

Figure 5. Global network of GPS stations that are routinely analyzed continuously at University of Nevada, Reno, using JPL's GIPSY-OASIS II software implementing the PPP method [*Zumberge et al.*, 1997] followed by the Ambizap method. Shown here, for example, are the 2568 baselines that were selected on 17 July 2007 by the procedure illustrated in Figure 3, then successfully bias fixed and network adjusted according to Figure 4. Only one path connects any pair of stations. Details of baselines in North America are shown in Figure 6.

independent of choice of the root station (in fact, alphabetical order is used).

[59] As a final note on software engineering, the network adjustment software ("netmrg") is recursive, where input arguments refer to each branch of the tree below as a new problem to solve. The recursive loop starts at the root station, calling new instances at each station down the tree. The loop is broken when reaching a leaf station of the tree, where the output solution simply equals the input solution. This is the first output filter solution. This information then feeds into the calling routine (next station up the tree), which then combines the information, and so on, until the calling program (for the root station) is reached. The smoother is basically the same recursive engine in reverse, writing output solutions before calling itself at each node. Recursion simplifies the code, as it does not need to model specific tree structure.

4. Testing

[60] This section presents the results of some basic tests, including validation of the baseline selection algorithm; timing of computation as a function of number of stations, and validation of $O(n)$ performance; validation that Ambizap gives positioning results that closely approximate full network ambiguity resolution; and verification that Ambizap improves positioning precision and accuracy, as should be the case if ambiguities are accurately resolved. The tests are intentionally not exhaustive, as ultimately the performance of Ambizap is better assessed by independent investigators for their specific geodetic problems at hand. The goal here is to provide evidence that both the underlying theory and practical implementation are basically sound.

4.1. Baseline Selection

[61] As a basic validation of the baseline selection algorithm (as explained section 3.2 and Figure 3), Figure 5 plots an example of all baselines selected on a typical (recent) day, in this case 17 July 2007. On this day, 2570 stations were processed, only one of which failed to be ambiguity resolved to the global network. Thus 2568 bias-fixed baselines are plotted here. By inspection, the resulting tree appears to closely approximate the EMST (section 2.5). In generally the tree will not be the EMST, as there are cases where ambiguity resolution fails, and another line of the Delauney triangulation is selected instead. Figure 6 shows in greater detail the selection of baselines in North America.

[62] Apart from validating the software, inspection of such plots reveals the weakest links in the global GPS network, corresponding to regions spanned by the longest baselines in the EMST. Such plots may assist in determining suitable locations of new stations to improve the probability of success of global ambiguity resolution. Such plots also indicate large regions where network ambiguity resolution is likely to be extremely robust, such as the contiguous United States, Europe, Japan, New Zealand, and South Africa. Therefore, sections 4.3 and 4.4 select North America as a region to test the improvement in positioning precision and accuracy that is attributable to Ambizap (following PPP).

4.2. Processing Time

[63] Run times for Ambizap were recorded (Figure 7) as a function of number of stations in the range $10 \leq n < 2000$. The computations were performed on a single 3-GHz Xeon CPU. For $n < 100$ the network tested was regional, in western North America. For $n > 1000$ the network is necessarily global (to find data from that many stations).

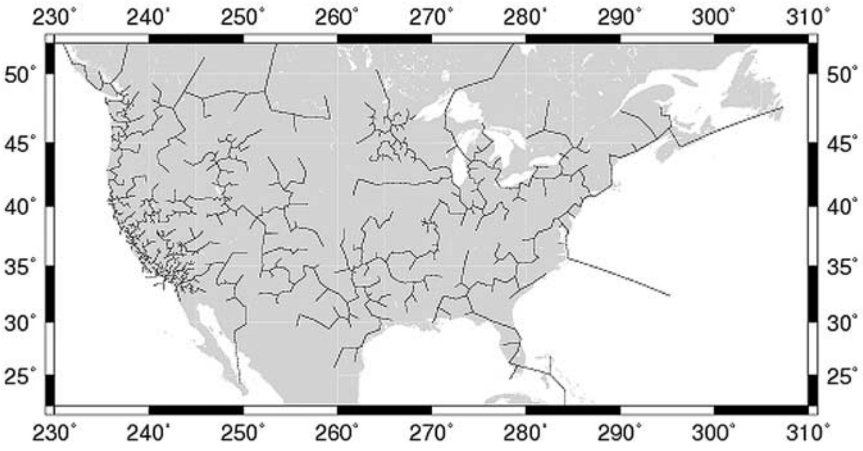

Figure 6. A zoomed view of Figure 5, showing details of 1554 stations and their selected baselines in North America. The dense cluster in the western United States is dominated by the EarthScope Plate Boundary Observatory.

[64] For comparison, run times were also computed for full network ambiguity resolution using modules distributed with the GIPSY-OASIS II software. These modules use core engine for bias fixing, known as Ambigon [*Blewitt*, 1989], which is the same engine at the core of Ambizap (for which it only operates on station pairs). Two implementations of full network ambiguity resolution were tested: (1) pure Ambigon, which implements full network bootstrapping over the entire network, and (2) Ambigon_p1 [*Hurst*, 2001], which is an iterative wrapper around Ambigon, operating on clusters of stations.

[65] The results (Figure 7) show that Ambizap performance is approximately $O(n)$, and so meets this key design specification. There appears to be slightly slower performance than $O(n)$ for $n < 100$ and slightly faster performance than $O(n)$ for $n > 100$. As discussed at the end of section 2.5, deviations from $O(n)$ can be expected depending on the failure rate of ambiguity resolution, in particular, the failure rate does tend to drop at high n for the global-scale network that was analyzed at $n > 100$. At the smallest values of n, the failure rate also drops, as the network becomes regional in scale. Thus the small deviation from $O(n)$ is caused by a slight hump in failure rates, causing more candidate baselines (in the Delauney triangulation) to be bias fixed. In any case, the deviation from $O(n)$ is rather small, and there is no hint of a transition to $O(n\log n)$ behavior at high n (tested to $n \approx 3000$).

[66] In terms of speed relative to PPP, Ambizap's processing time is ~5 s per station, as compared to PPP's ~12 s per station. Thus Ambizap adds only a fraction of overhead to PPP processing time. Moreover, the solutions are block diagonal, and so can be written out into individual station files in exactly the same format as for PPP. In contrast full network ambiguity resolution by Ambigon displays $O(n^4)$ behavior, making it impractical to process (and test) networks of $n > 100$. The clustered bootstrapping method of Ambigon_p1 clearly outperforms Ambigon on its own, but nevertheless shows the same $O(n^4)$ behavior.

[67] Ambizap is now routinely applied at UNR for the analysis of ~3000 stations from the global network (at the time of writing, mid-2008), with PPP initially applied using JPL's GGN products. Using this method, UNR has processed most of the world's GPS geodetic data back to 1994 (Figure 1) when GGN/IGS products became available [*Beutler et al.*, 1994], with ~10 days of data processing on a ~40 × 3 GHz Xeon CPU cluster (using custom cluster

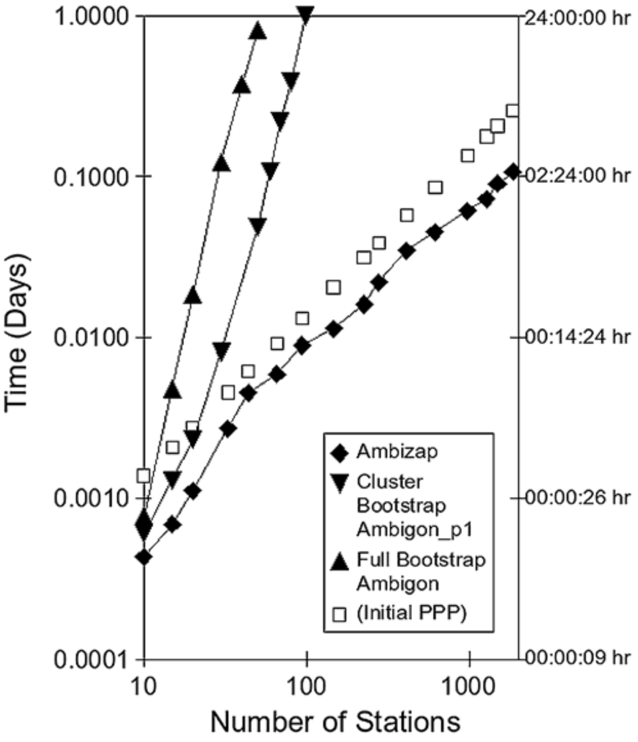

Figure 7. Log-log plot of run times for Ambizap versus number of stations up to n = 2000, computed on a single 3-GHz Xeon CPU. Ambizap demonstrates behavior consistent with $O(n)$ (slope of unity). Shown for comparison are full network bootstrapping techniques showing $O(n^4)$ behavior, and the initial PPP computation, which is exactly $O(n)$ (by definition). For convenience, the time scale on the right is in the format hours:minutes:seconds (hh:mm:ss).

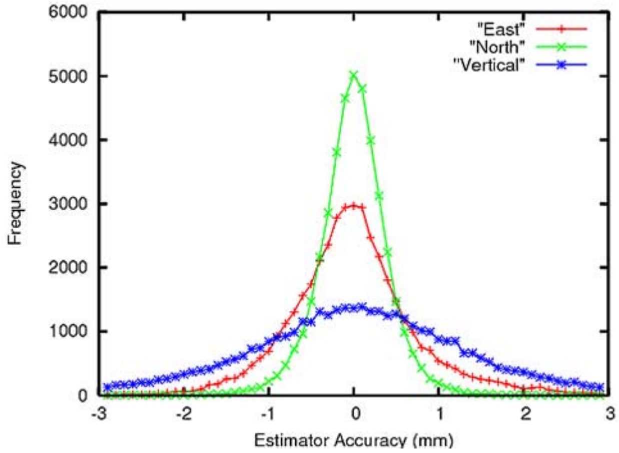

Figure 8. Frequency plot of estimator accuracy, defined as the difference between coordinate estimates produced by Ambizap and Ambigon in full network (bootstrapping) mode.

software with a passive server/demanding client model). Most of the sites analyzed at UNR (~1600) are in North America (Figure 6).

4.3. Estimator Accuracy

[68] Estimator accuracy here is defined as the difference between coordinate estimates produced on the one hand by the Ambizap algorithm developed here and on the other hand by the Ambigon_p1 cluster bootstrapping algorithm. (Note that Ambigon and Ambigon_p1 are both rigorous full network algorithms and so give identical results). Estimator accuracy is not a measure of how accurate the estimated positions are (which is discussed in sections 4.4 and 4.5). Rather it is a measure of how well the Ambizap algorithm reproduces the results of full network ambiguity resolution.

[69] Estimator accuracy was tested by processing data from a regional network of 30 stations in the southern BARGEN array of western North America in the time frame 2003.0–2006.9. This gave 41,625 daily station coordinate differences for each of the east, north, and vertical component. A frequency plot of the resulting differences is presented in Figure 8. The estimator accuracy has an RMS (about zero) of 0.43 mm (north), 0.76 mm (east), and 1.67 mm (vertical). This is to be compared with RMS between Ambizap and the initial daily PPP solutions of 0.56 mm (north), 3.51 mm (east), and 2.17 mm (vertical). The variance ratio in the east component is 21.3, indicating that the difference between Ambizap solutions and Ambigon solution is negligible when compared to the difference with the initial PPP solution. Therefore Ambizap proves to meet the key design requirement of closely replicating the Ambigon solution with orders of magnitude less processing time.

4.4. Positioning Precision

[70] If ambiguity resolution is working correctly, we should expect (on the basis of covariance analysis and previous tests [e.g., *Blewitt*, 1989]) significant improvements to coordinate precision in the east component for stations at midlatitudes. Positioning precision was assessed by comparing results from Ambizap versus initial PPP of the long-term daily coordinate repeatability of stations with long, unbroken time series in the dense North American cluster.

[71] Figure 9 shows a typical example of detrended coordinate time series before and after the application of Ambizap to PPP solutions in a realization of a stable North America reference frame. The mean repeatability (RMS residual) is significantly improved in the east component from 2.6 mm before ambiguity resolution, to 1.6 mm after ambiguity resolution. This indicates that the initial estimation of biases as real-valued parameters adds noise to the time series at the level of 2.0 mm, which is removed by ambiguity resolution. It is also evidence that Ambizap is working as intended.

4.5. Geodetic Accuracy: Velocities in "Stable North America"

[72] If ambiguity resolution is working correctly, we should also expect positioning accuracy to improve, as has been demonstrated by comparison with independent positioning techniques [*Blewitt*, 1989]. GPS accuracy has become difficult to test due to the lack of colocated independent techniques of significantly higher accuracy. Another way to assess geodetic accuracy is to compare station velocities (fitting the position time series) to a simple geophysical model [*Davis et al.*, 2003].

[73] Here the accuracy of site velocities is estimated by assuming zero motion (with respect to a nonrotating reference frame) for stations located in the stable plate interior of North America, far from tectonic effects and regions known to be deforming from glacial isostatic adjustment [*Sella et al.*, 2007; *Calais et al.*, 2006]. The analysis imposes the additional stringent criterion that the sites must have been continuously operating during the years 2000–2007, with no discontinuities in the time series (as a result of equipment configuration changes). The results show that the RMS of the east velocities is 0.9 mm/a in stable North America before running Ambizap, reducing to 0.7 mm after running Ambizap. (The RMS north and vertical velocities do not

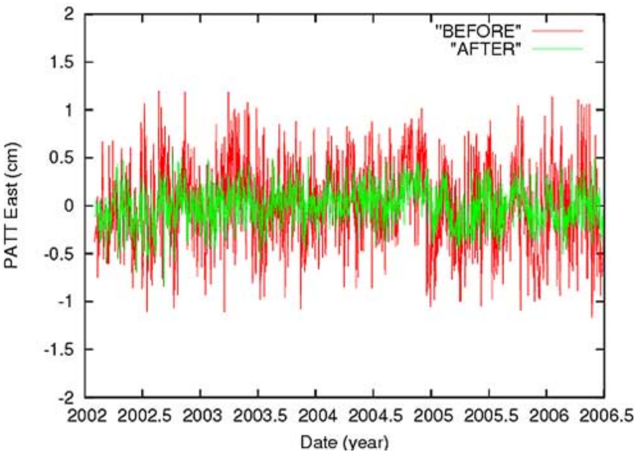

Figure 9. A typical example of coordinate time series before and after the application of Ambizap to PPP solutions, for station PATT in the east component (detrended).

change.) This is independent evidence that Ambizap is improving accuracy.

5. Conclusions

[74] A new algorithm known as Ambizap has been demonstrated for the bias fixing of continuous GPS networks. Ambizap is based on a new theoretical foundation of fixed point theorems, stochastic properties of large bias-fixed networks, and the application of existing theory on minimum spanning trees.

[75] Ambizap demonstrates submillimeter agreement with classical full network algorithms, but computes the result as $O(n)$ rather than $O(n^4)$, which is currently the case for the GIPSY OASIS II software in common use for high-precision geodesy. For example, a 2000-station network can be bias fixed in 2.5 h on a 3-GHz single CPU computer. This allows for the rapid ambiguity resolution of global networks of several thousands of stations, whereas current $O(n^4)$ algorithms are limited to networks of <100 stations, requiring ad hoc methods to combine smaller network solutions. In contrast, Ambizap produces a unique solution without any assignment of stations to networks (indeed, there is only one network), which radically facilitates data management. Moreover, the addition of extra station data to a network solution is very straightforward and fast, thus removing the operational burden of waiting until all potential data are in hand prior to performing a network analysis.

[76] Application of the Ambizap algorithm will greatly assist analysis of crustal movement in regions such as the western North America, which have dense overlapping GPS networks. For example, a network solution from 1 day of the ~1000 station Plate Boundary Observatory can be produced from RINEX files in about 7 min on a 40-CPU cluster (4.5 min PPP + 2.5 min Ambizap). Note that, on theoretical grounds, Ambizap is not optimally suited for the analysis of GPS campaign data.

[77] What is perhaps most significant is that the improvement in precision and accuracy allows for finer temporal resolution on station displacements, which is important for resolving transient signals, such as postseismic deformation. Since Ambizap can produce a unique dense network solution for the entire globe, this enhances the potential to detect transient signals from local to global scales, without need to predefine the networks subject to investigations. This should therefore enhance paths to future discovery in the application of geodesy to tectonophysics and geodynamics.

[78] **Acknowledgments.** The author thanks the Jet Propulsion Laboratory, Caltech, for making available the GIPSY OASIS II software, together with precise GPS orbit and clock solutions derived from the International GNSS Service (IGS) network. Yoaz Bar Sever compiled larger executable versions of Ambigon for timing comparisons presented here. The IGS and its participating regional networks provided GPS data used in this paper. Users of Ambizap made useful suggestions to improve the prototype version of Ambizap; in particular, Mark Simons suggested using the Delauney triangulation and the software library STRIPAK by Robert Renka, University of North Texas. The manuscript was improved by constructive suggestions of Kristine Larson and an anonymous reviewer. This research was supported in part by the DOE Yucca Mountain Project/NSHE Cooperative Agreement DE-FC28-04RW12232, task ORD-FY04-003, NASA/JPL grant 1324361, NASA IDS grant NNG04G099G, and NSF Tectonics/EPSCOR grant EAR-0610031.

References

Aurenhammer, F. (1991), Voronoi diagrams—A study of a fundamental geometric data structure, *ACM Comput. Surv.*, *23*, 345–405, doi:10.1145/116873.116880.

Beutler, G., I. I. Mueller, and R. E. Neilan (1994), The International GPS Service for Geodynamics: Development and start of official service on January 1, 1994, *Bull. Geod.*, *68*, 39–70.

Blewitt, G. (1989), Carrier phase ambiguity resolution for the Global Positioning System applied to geodetic baselines up to 2000 km, *J. Geophys. Res.*, *94*, 10,187–10,283, doi:10.1029/JB094iB08p10187.

Blewitt, G. (2007), GPS and space based geodetic methods, in *Treatise on Geophysics*, vol. 3, edited by T. Herring, pp. 351–390, Academic, Oxford, U.K.

Calais, E., J. Y. Han, C. DeMets, and J. M. Nocquet (2006), Deformation of the North American plate interior from a decade of continuous GPS measurements, *J. Geophys. Res.*, *111*, B06402, doi:10.1029/2005JB004253.

Davis, J. L., R. A. Bennett, and B. P. Wernicke (2003), Assessment of GPS velocity accuracy for the Basin and Range Geodetic Network (BARGEN), *Geophys. Res. Lett.*, *30*(7), 1411, doi:10.1029/2003GL016961.

Dong, D.-N., and Y. Bock (1989), GPS network analysis with phase ambiguity resolution applied to crustal deformation studies in California, *J. Geophys. Res.*, *94*, 3949–3966, doi:10.1029/JB094iB04p03949.

Dow, J. M., R. E. Neilan, and G. Gendt (2005), The International GPS Service (IGS): Celebrating the 10th anniversary and looking to the next decade, *Adv. Space Res.*, *36*(3), 320–326, doi:10.1016/j.asr.2005.05.125.

Galil, Z., and G. F. Italiano (1991), Data structure and algorithms for disjoint set union problems, *ACM Comput. Surv.*, *23*(3), 319–344, doi:10.1145/116873.116878.

Hurst, K. (2001), The JPL SCIGN-3.0.0 analysis—Automation and results (2001), *Eos Trans. AGU*, *82*(47), Fall Meet. Suppl., Abstract G41A-0205.

Kruskal, J. B. (1956), On the shortest spanning subtree of a graph and the traveling salesman problem, *Proc. Am. Math. Soc.*, *7*(1), 48–50, doi:10.2307/2033241.

Melbourne, W. G. (1985), The case for ranging in GPS based geodetic systems, paper presented at the First Symposium on Precise Positioning with the Global Positioning System, Positioning with GPS-1985, U. S. Dep. of Commer., Rockville, Md.

Moore, A. W. (2007), The International GNSS Service: Any questions?, *GPS World*, *18*(1), 58–64.

Renka, R. (1997), Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere, *Trans. Math. Software*, *23*(3), 416–434, doi:10.1145/275323.275329.

Sella, G. F., S. Stein, T. H. Dixon, M. Craymer, T. S. James, S. Mazzotti, and R. K. Dokka (2007), Observation of glacial isostatic adjustment in "stable" North America with GPS, *Geophys. Res. Lett.*, *34*, L02306, doi:10.1029/2006GL027081.

Shashkin, Y. A. (1991), *Fixed Points*, 370 pp., Am. Math. Soc., Providence, R. I.

Teunissen, P. J. G. (1995), The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation, *J. Geod.*, *70*, 65–82, doi:10.1007/BF00863419.

Wübbena, G. (1985), Software developments for geodetic positioning with GPS using TI-code and carrier measurements, paper presented at the First Symposium on Precise Positioning with the Global Positioning System, Positioning with GPS-1985, U. S. Dep. of Commer., Rockville, Md.

Zumberge, J. F., M. B. Heflin, D. C. Jefferson, M. M. Watkins, and F. H. Webb (1997), Precise point positioning for the efficient and robust analysis of GPS data from large networks, *J. Geophys. Res.*, *102*, 5005–5017, doi:10.1029/96JB03860.

G. Blewitt, Nevada Bureau of Mines and Geology, Mail Stop 178, 1664 N. Virginia Street, University of Nevada, Reno, NV 89557-0178, USA. (gblewitt@unr.edu)