Computation and Performance Studies
Volume 1 · Number 1 · 2026

A Quaternion Sign Can Change a Naive Midpoint Without Changing Either Endpoint Pose

Cameron Porter[1] · Jack Brooks[2] · James Harrison[3]

[1] Canada · [2] Germany · [3] United Kingdom

Original publication: October 28, 2026 | Article 654

Abstract

Two quaternions with opposite signs encode the same rotation, yet their component average can behave very differently. We examined 615,364 short-interval pose pairs in 52 TUM RGB-D files. Native endpoint signs caused 778 naive normalized midpoints to be undefined or more than one degree from the shortest-path midpoint. The equal-file failure fraction was 0.1732%. Aligning endpoint hemispheres before averaging reduced the maximum observed midpoint discrepancy to numerical precision. This is a representation-consistency result for two-pose midpoints, not a test of physical intermediate orientation or a learned motion model.

Representation audit of native camera quaternions. The deliberately naive midpoint is our comparison rule; native sign choices are valid rotation representations and are not labelled dataset errors.

A Quaternion Sign Can Change a Naive Midpoint Without Changing Either Endpoint Pose

> Representation audit of native camera quaternions. The deliberately naive midpoint is our comparison rule; native sign choices are valid rotation representations and are not labelled dataset errors.

Abstract

Two quaternions with opposite signs encode the same rotation, yet their component average can behave very differently. We examined 615,364 short-interval pose pairs in 52 TUM RGB-D files. Native endpoint signs caused 778 naive normalized midpoints to be undefined or more than one degree from the shortest-path midpoint. The equal-file failure fraction was 0.1732%. Aligning endpoint hemispheres before averaging reduced the maximum observed midpoint discrepancy to numerical precision. This is a representation-consistency result for two-pose midpoints, not a test of physical intermediate orientation or a learned motion model.

Why a small representation choice reaches a motion calculation

A camera pose pipeline can read every quaternion field correctly and still combine two orientations incorrectly. The difficulty is that a unit quaternion and its negative represent the same physical rotation. Averaging four stored components therefore differs from averaging the rotations they represent. Normalizing the resulting vector fixes its length but does not by itself remove dependence on the chosen signs.

We ask how often a simple normalized component midpoint disagrees with a sign-consistent shortest-path midpoint on native short-interval camera records. No artificial sign reversals are inserted into the source sequence for the primary comparison. The experiment uses the representations already present in the acquired files and evaluates two explicitly defined ways of combining them.

Meng, 2026 studies disentangling camera and subject motion with a geometry-semantics attention mechanism. We do not reproduce that network or suggest that it uses the naive midpoint considered here. The connection is a basic condition for interpreting camera motion: a numerical operation should respect equivalence in its pose representation before its output is read as a physical change.

The comparison is an implementation audit, not a new rotation-averaging theory. Quaternion averaging literature explains the sign-invariance problem directly (Markley et al., 2007), and broader rotation-averaging work distinguishes different averaging objectives (Hartley et al., 2013). We use the narrow two-pose, equal-weight midpoint case to measure how often an elementary failure can occur in a concrete public reference inventory.

Native orientation records and the selected intervals

TUM RGB-D provides the recorded camera trajectories used in the audit (Sturm et al., 2012). The official format specifies the color camera's optical-center orientation as a quaternion ordered by three vector components followed by its scalar component (TUM Computer Vision Group, 2026). Position and orientation share the motion-capture world convention. We do not substitute an Euler-angle interpretation for those four values.

The acquired text subset contains 53 public files, including one comment-only source. After excluding all rows at conflicting timestamps in a separate documented analysis copy, 52 files contain 615,775 ordered

numerical poses. The raw files remain available. This study uses the same curated reference inventory as other local trajectory audits and does not claim an independent acquisition replication.

We retain adjacent pose pairs whose timestamps differ by at most fifty milliseconds. This removes intervals for which a short-gap midpoint would span a wider missing-reference period. The resulting set has 615,364 pairs. The cutoff defines the inspected cohort; it is not an assertion that every physical camera motion between those endpoints follows a constant angular rate.

Source quaternions are normalized before the comparison. That removes small text-rounding deviations from unit length and makes the two formulas operate on the same unit sphere. It does not choose their signs or modify the rotations. Matrix representations calculated from a normalized quaternion and its negative agree exactly at the stored execution precision in our sign-reversal check.

The camera recordings include motion and calibration roles, with related devices and settings. Other resources such as KITTI, EuRoC and TUM VI have different recording designs (Geiger et al., 2012; Burri et al., 2016; Schubert et al., 2018). Their records do not increase our sample size. The 52-file summary belongs to this acquired collection rather than to all possible camera-orientation streams.

Three midpoint calculations

The naive rule adds the two unit quaternion vectors and normalizes their sum. If its norm is at most 1e-12, we record an undefined midpoint. Otherwise, we convert that normalized vector back to a rotation. The rule is deliberately transparent, making its representation dependence testable without introducing another estimator or tuning parameter.

The sign-aware rule first checks the endpoint dot product. If it is negative, the second quaternion is negated before the vectors are added and normalized. This places the two endpoint representations in a common hemisphere. Negation leaves the second endpoint's rotation unchanged while changing which arc the component sum follows.

For the reference midpoint, we compute the relative rotation from the first endpoint to the second, take half its shortest rotation vector and compose that half rotation with the first endpoint. The exponential-map representation provides relevant geometric context (Grassia, 1998), while rotation-metric analysis explains why the disagreement should be measured as a rotation rather than a raw vector difference (Huynh, 2009).

At equal weight one half, the hemisphere-aligned normalized sum agrees with the shortest geodesic midpoint under the stated convention. This identity should not be extended without qualification to arbitrary interpolation fractions or weighted averages of many rotations. General quaternion interpolation and averaging require their own definitions; the present calculation does not implement a general optimal averaging procedure.

Shoemake's interpolation work and the versioned SciPy documentation supply the interpolation context (Shoemake, 1985; SciPy developers, 2026). The executed implementation uses numerical rotation composition in the local scientific environment (Virtanen et al., 2020). It does not evaluate a learned orientation predictor, a smoothing spline or the hidden physical trajectory between the two measurements.

Defining a visible disagreement

We measure the angular magnitude of the relative rotation between a computed midpoint and the geodesic reference, converting that angle to degrees. A naive midpoint is classified as failed if it is undefined or its angular discrepancy exceeds one degree. Undefined sums remain in the failure denominator instead of disappearing through a numerical-value filter.

One degree is an operational reporting threshold, not a validated tolerance for a camera-transfer application. The numerical discrepancy of the sign-aware rule is reported separately. This keeps a finite-precision consistency check distinct from an application decision about how much orientation error is acceptable.

Rigid transformation references clarify why rotation representation and physical orientation are different levels of description (Barfoot, 2017; Blanco-Claraco, 2021). Multi-view geometry then uses those rotations in projections and transformations (Hartley and Zisserman, 2004). A midpoint artifact can enter such a calculation even though both endpoint matrices represent the intended poses correctly.

The native negative-dot pairs are not removed as bad observations. Either endpoint sign is valid. Dropping one would discard a legitimate pose because a particular arithmetic formula mishandles its representation. Robust location methods address a different concern about unusual observations (Huber, 1964); we do not fit a robust estimator or treat valid quaternion signs as physical outliers.

A global sign convention is not the same as local sign alignment. For example, requiring every quaternion's scalar component to be nonnegative selects one representative for most rotations. Near a rotation whose scalar component crosses zero, that canonical choice can still produce a discontinuity in the stored vector. The physical rotation need not jump at the same place. Local endpoint alignment addresses the particular pair being combined rather than assuming that a global component convention guarantees a continuous vector stream.

There is also a useful distinction in how an implementation is tested. Negating every quaternion in a sequence simultaneously leaves each naive normalized sum globally negated, which encodes the same midpoint rotation as before. A test using only that common sign reversal could pass while missing the failure under an independent reversal of one endpoint. Equivalent representations must therefore be varied at the level relevant to the operation. The native pairs here supply concrete instances of that issue without altering their endpoint information.

A file-level failure estimate

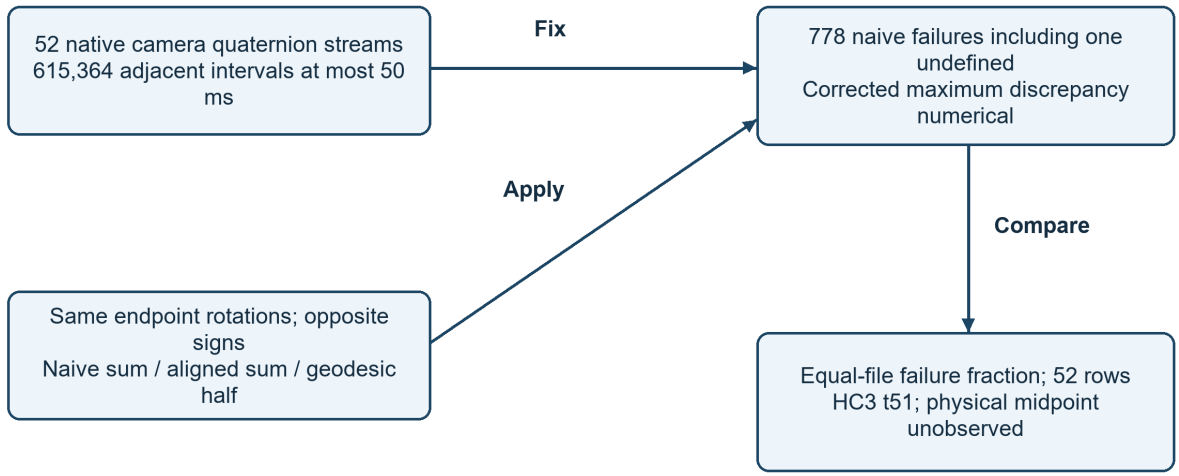


Figure 1. *Native endpoint quaternions define unchanged rotations. Naive and hemisphere-aligned normalized sums are compared with a geodesic midpoint; undefined results and angular failures are counted within file before aggregation.*

For each file, we divide its naive midpoint failures by its retained short-interval pair count. The regression estimates the mean of these 52 fractions using an intercept, HC3 covariance and t51 intervals. No covariates, file effects or pair-level independence assumption are used. The coefficient is equal-file weighted, not weighted by how many intervals a file contains.

Heteroskedasticity-consistent variance is a working description of variation across those fractions (White, 1980). Shared acquisition conditions can create dependence not handled by that calculation. Clustered inference would need an appropriate group structure and enough independent groups (Cameron and Miller, 2015). We do not interpret the interval as a guarantee for another camera or pose-storage system.

Table 1. *Equal-file naive midpoint failure fraction, N=52, intercept only, HC3 standard error and t51 confidence interval. R-squared is numerically zero. The failure event includes undefined component sums as well as discrepancies above the operational angular threshold.*

Model	Term	Estimate (SE)	95% CI	N
M1	Mean outcome/contrast	0.001732 (0.000344)	[0.001042, 0.002421]	52
M1: fit	R-squared = 0.0000	-	-	52

Model labels: M1 = Equal-file naive midpoint failure fraction.

Forty files contain at least one negative-dot endpoint pair. There are 778 such pairs in total, and the naive rule records 778 failures. One has an undefined raw midpoint under the norm threshold. The other failures have defined quaternion vectors whose rotations disagree with the selected shortest-path midpoint. This count does not mean that the source files contain 778 incorrect endpoint orientations.

The equal-file mean failure fraction is 0.00173163, standard error 0.00034360 and interval [0.00104182, 0.00242144]. In percentage terms, the point estimate is approximately 0.1732%. Pooling intervals instead gives 0.00126429, about 0.1264%, because long files receive more weight. The denominators describe different aggregate targets.

The maximum sign-aware midpoint discrepancy is approximately 4.28e-14 degrees. Its numerical agreement with the reference is expected from the midpoint identity and supports the implemented convention. It is not a zero-error estimate of the true camera orientation, nor does the absence of an observed numerical failure establish population coverage for future data.

The endpoint sign-reversal matrix check has maximum difference zero at the execution precision. That verifies the premise that the input rotations are unchanged by sign reversal. Combined with the different naive midpoints, it directly shows how a representation-dependent operation can create a disagreement without changing the endpoint pose information.

Physical motion and numerical disagreement remain different targets

A pair can legitimately describe different physical orientations at its two endpoints while allowing either quaternion sign for each. The sign-aware rule chooses the shortest rotational path for its midpoint. Whether the actual camera followed that path between observations is a separate measurement question. We do not observe intermediate physical orientation merely by calculating an interpolant.

Trajectory-evaluation work makes temporal and spatial correspondence part of a coherent comparison (Zhang and Scaramuzza, 2018). Here the endpoint times are fixed and short-gap eligibility is common. The remaining discrepancy is introduced by the combination rule. It should not be relabelled as a recovered subject movement, a camera-tracking error or an identity-consistency failure in generated video.

Measurement-error frameworks help separate such operational outputs from the physical constructs they are intended to describe (Carroll et al., 2006). A scalar degree error is also different from a quaternion Euclidean distance or a pixel reprojection error (Stevens, 1946). We retain the rotation metric and its unit throughout, rather than compare these quantities as if they measured the same scale.

Numerical treatment near a small vector sum deserves attention (Higham, 2002; Goldberg, 1991). Our norm threshold prevents an unstable normalization from producing an arbitrary apparently valid rotation. It does not hide the case: undefined output is counted as a failure and its source location is retained. An implementation that silently replaces it with an identity rotation would introduce another untested policy.

What follows for geometric pipelines

Simulation resources such as CARLA and Virtual KITTI 2 can offer controlled geometric annotations (Dosovitskiy et al., 2017; Yohann et al., 2020), but those coordinates still require operations that respect their representation. The sign issue is mathematical rather than a defect unique to a real recording. We do not run simulations or infer that their software uses the naive formula from this reference-file audit.

A practical orientation pipeline should state quaternion ordering, normalization, interpolation convention and the metric used to check results. The current two-pose calculation illustrates why a test should include equivalent endpoint representations. It does not prescribe a complete filtering or estimation method for a noisy sequence, where uncertainty, dynamics and many-pose averaging would require additional design choices.

Raw component differences deserve similar interpretation. Two nearby physical orientations can have quaternion representatives whose Euclidean difference is close to two when their signs oppose. Thresholding that vector jump could flag a discontinuity in representation rather than a large camera rotation. We do not compute a detector's false-alarm rate from this observation; it identifies another operation whose invariance would need explicit testing.

For a learned pipeline that consumes quaternion components directly, a corresponding follow-up could feed equivalent endpoint signs and compare outputs under a fixed model. Such a test would need the actual model and a declared output metric. Our arithmetic comparison cannot predict whether a network's preprocessing already resolves the sign ambiguity, learns robustness to it or passes it into a downstream prediction.

Influential files and their sign-switch patterns remain inspectable in the saved summaries (Cook, 1977). Broader validation would need the intended source conventions and acquisition boundary (Roberts et al., 2017). Repeating the same native file through another formula does not create another independent camera experiment, even when the numerical comparison is fully deterministic.

The artifacts retain file identities, pair counts and example failure locations, with separate undefined and angular-discrepancy information. Dataset documentation records those meanings (Gebru et al., 2021), while provenance links the audit to the original and curated pose archives (Wilkinson et al., 2016). The result isolates a small but concrete class of numerical midpoint failures that can be prevented by respecting a rotation's sign equivalence before interpreting the output as motion.

References

1. Zhang, Zichao; Scaramuzza, Davide. (2018). A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/IROS.2018.8593941.
2. Wilkinson, Mark D.; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Bonino; Bourne, Philip E.; Bouwman, Jildau; Brookes, Anthony J.; Clark, Tim; Crosas, Mercè; Dillo, Ingrid; Dumon, Olivier; Edmunds, Scott; Evelo, Chris T.; Finkers, Richard; Gonzalez-Beltran, Alejandra; Gray, Alasdair J.G.; Groth, Paul; Goble, Carole; Grethe, Jeffrey S.; Heringa, Jaap; 't Hoen, Peter A.C; Hooft, Rob; Kuhn, Tobias; Kok, Ruben; Kok, Joost; Lusher, Scott J.; Martone, Maryann E.; Mons, Albert; Packer, Abel L.; Persson, Bengt; Rocca-Serra, Philippe; Roos, Marco; van Schaik, Rene; Sansone, Susanna-Assunta; Schultes, Erik; Sengstag, Thierry; Slater, Ted; Strawn, George; Swertz, Morris A.; Thompson, Mark; van der Lei, Johan; van Mulligen, Erik; Velterop, Jan; Waagmeester, Andra; Wittenburg, Peter; Wolstencroft, Katherine; Zhao, Jun; Mons, Barend. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. https://doi.org/10.1038/sdata.2016.18.
3. Gebru, Timnit, et al. "Datasheets for datasets" *Communications of the ACM*, 2021. https://doi.org/10.1145/3458723.
4. Dosovitskiy, Alexey; Ros, German; Codevilla, Felipe; Lopez, Antonio; Koltun, Vladlen. (2017). CARLA: An Open Urban Driving Simulator. arXiv:1711.03938. https://arxiv.org/abs/1711.03938.
5. Stevens, S. S.. (1946). On the Theory of Scales of Measurement. Science. https://doi.org/10.1126/science.103.2684.677.
6. Cook, R. Dennis. (1977). Detection of Influential Observation in Linear Regression. Technometrics. https://doi.org/10.1080/00401706.1977.10489493.
7. Meng, Zijie. (2026). OrthoMotion:Disentangling Camera and Subject Motion via Geometry Semantics Orthogonal Attention. arXiv:2606.22835. https://arxiv.org/abs/2606.22835.
8. Burri, Michael; Nikolic, Janosch; Gohl, Pascal; Schneider, Thomas; Rehder, Joern; Omari, Sammy; Achtelik, Markus W; Siegwart, Roland. (2016). The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research. https://doi.org/10.1177/0278364915620033.

9. Virtanen, Pauli; Gommers, Ralf; Oliphant, Travis E.; Haberland, Matt; Reddy, Tyler; Cournapeau, David; Burovski, Evgeni; Peterson, Pearu; Weckesser, Warren; Bright, Jonathan; van der Walt, Stéfan J.; Brett, Matthew; Wilson, Joshua; Millman, K. Jarrod; Mayorov, Nikolay; Nelson, Andrew R. J.; Jones, Eric; Kern, Robert; Larson, Eric; Carey, C J; Polat, İlhan; Feng, Yu; Moore, Eric W.; VanderPlas, Jake; Laxalde, Denis; Perktold, Josef; Cimrman, Robert; Henriksen, Ian; Quintero, E. A.; Harris, Charles R.; Archibald, Anne M.; Ribeiro, Antônio H.; Pedregosa, Fabian; van Mulbregt, Paul; SciPy 1.0 Contributors; Vijaykumar, Aditya; Bardelli, Alessandro Pietro; Rothberg, Alex; Hilboll, Andreas; Kloeckner, Andreas; Scopatz, Anthony; Lee, Antony; Rokem, Ariel; Woods, C. Nathan; Fulton, Chad; Masson, Charles; Häggström, Christian; Fitzgerald, Clark; Nicholson, David A.; Hagen, David R.; Pasechnik, Dmitrii V.; Olivetti, Emanuele; Martin, Eric; Wieser, Eric; Silva, Fabrice; Lenders, Felix; Wilhelm, Florian; Young, G.; Price, Gavin A.; Ingold, Gert-Ludwig; Allen, Gregory E.; Lee, Gregory R.; Audren, Hervé; Probst, Irvin; Dietrich, Jörg P.; Silterra, Jacob; Webber, James T; Slavič, Janko; Nothman, Joel; Buchner, Johannes; Kulick, Johannes; Schönberger, Johannes L.; de Miranda Cardoso, José Vinícius; Reimer, Joscha; Harrington, Joseph; Rodríguez, Juan Luis Cano; Nunez-Iglesias, Juan; Kuczynski, Justin; Tritz, Kevin; Thoma, Martin; Newville, Matthew; Kümmerer, Matthias; Bolingbroke, Maximilian; Tartre, Michael; Pak, Mikhail; Smith, Nathaniel J.; Nowaczyk, Nikolai; Shebanov, Nikolay; Pavlyk, Oleksandr; Brodtkorb, Per A.; Lee, Perry; McGibbon, Robert T.; Feldbauer, Roman; Lewis, Sam; Tygier, Sam; Sievert, Scott; Vigna, Sebastiano; Peterson, Stefan; More, Surhud; Pudlik, Tadeusz; Oshima, Takuya; Pingel, Thomas J.; Robitaille, Thomas P.; Spura, Thomas; Jones, Thouis R.; Cera, Tim; Leslie, Tim; Zito, Tiziano; Krauss, Tom; Upadhyay, Utkarsh; Halchenko, Yaroslav O.; Vázquez-Baeza, Yoshiki. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods. https://doi.org/10.1038/s41592-019-0686-2.

10. SciPy developers. (2026). scipy.spatial.transform.Slerp [API documentation; snapshot accessed 7 November 2026]. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.transform.Slerp.html.

11. Schubert, David; Goll, Thore; Demmel, Nikolaus; Usenko, Vladyslav; Stückler, Jörg; Cremers, Daniel. (2018). The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. arXiv:1804.06120. https://arxiv.org/abs/1804.06120.

12. Blanco-Claraco, José Luis. (2021). A tutorial on SE(3) transformation parameterizations and on-manifold optimization. arXiv:2103.15980; arXivversionyear, originaltechnicalreport2010. https://arxiv.org/abs/2103.15980.

13. Hartley, Richard; Trumpf, Jochen; Dai, Yuchao; Li, Hongdong. (2013). Rotation Averaging. International Journal of Computer Vision. https://doi.org/10.1007/s11263-012-0601-0.

14. Cameron, A. Colin, and Douglas L. Miller. “A Practitioner’s Guide to Cluster-Robust Inference.” *Journal of Human Resources*, vol. 50, no. 2, 2015, pp. 317-372. https://doi.org/10.3368/jhr.50.2.317.

15. Yohann, Cabon,; Naila, Murray,; Martin, Humenberger,. (2020). Virtual KITTI 2. https://arxiv.org/abs/2001.10773.

16. Shoemake, Ken. (1985). Animating rotation with quaternion curves. Proceedings of the 12th annual conference on Computer graphics and interactive techniques - SIGGRAPH '85. https://doi.org/10.1145/325334.325242.

17. TUM Computer Vision Group. (2026). RGB-D SLAM Dataset: File Formats [Official documentation; accessed8November2026, captureyearnotbenchmarkyear]. https://cvg.cit.tum.de/data/datasets/rgbd-dataset/file_formats.

18. Geiger, A.; Lenz, P.; Urtasun, R.. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. 2012 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2012.6248074.

19. Huber, Peter J.. (1964). Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics. https://doi.org/10.1214/aoms/1177703732.

20. Carroll, Raymond J., David Ruppert, Leonard A. Stefanski, and Ciprian M. Crainiceanu. “Measurement Error in Nonlinear Models.” 2006. https://doi.org/10.1201/9781420010138.

21. Hartley, Richard; Zisserman, Andrew. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press. https://doi.org/10.1017/cbo9780511811685.

22. Sturm, Jürgen; Engelhard, Nikolas; Endres, Felix; Burgard, Wolfram; Cremers, Daniel. (2012). A benchmark for the evaluation of RGB-D SLAM systems. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. https://doi.org/10.1109/IROS.2012.6385773.

23. Markley, F. Landis; Cheng, Yang; Crassidis, John L.; Oshman, Yaakov. (2007). Averaging Quaternions. Journal of Guidance, Control, and Dynamics. https://doi.org/10.2514/1.28949.

24. Barfoot, Timothy D.. (2017). State Estimation for Robotics. Cambridge University Press, first edition. https://doi.org/10.1017/9781316671528.

25. Goldberg, David. (1991). What every computer scientist should know about floating-point arithmetic. ACM Computing Surveys. https://doi.org/10.1145/103162.103163.

26. White, Halbert. “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity.” *Econometrica*, vol. 48, no. 4, 1980, pp. 817-838. https://doi.org/10.2307/1912934.

27. Grassia, F. Sebastian. (1998). Practical Parameterization of Rotations Using the Exponential Map. Journal of Graphics Tools. https://doi.org/10.1080/10867651.1998.10487493.

28. Higham, Nicholas J.. (2002). Accuracy and Stability of Numerical Algorithms: Second Edition. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718027.

29. Huynh, Du Q.. (2009). Metrics for 3D Rotations: Comparison and Analysis. Journal of Mathematical Imaging and Vision. https://doi.org/10.1007/s10851-009-0161-2.

30. Roberts, David R., et al. “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure.” *Ecography*, vol. 40, no. 8, 2017, pp. 913-929. https://doi.org/10.1111/ecog.02881.