Last Supper Supplement

Artist's Statement

The twelve apostles form a six-module, twelve-strut tensegrity ring. Each apostle is a strut. The ring is a closed string of six T3 prisms, but it is not rigid: each shared strut acts as a hinge, so the whole remains a linkage capable of flexing. The community is complete and interdependent, yet structurally unable to hold a fixed form on its own.

Six apostles are shared struts, belonging to two prisms at once; six are open struts, belonging to only one. Judas is a shared strut: the ring closes through him, and his removal would disrupt two prisms rather than one. John is an open strut placed nearest Jesus and oriented toward him. The two apostles seated closest to Christ in Leonardo da Vinci's Last Supper therefore occupy opposite structural conditions here. Peter, Andrew, James the Greater, Matthew, and Philip form the remaining shared members; the questioners, the doubter, and the zealot occupy open positions.

Jesus stands apart from the ring. His incomplete tensegrity components place him within the Unmoored series, but here he is oriented away from a former structure and toward another. In his hand is a complete T3 prism representing the Trinity. In contrast to the flexible apostolic linkage, this small prism is structurally resolved. The community he leaves is a linkage; what he holds is the kind of structural completeness it lacks.

The nodes surrounding Jesus and those of the apostolic ring share a single color, suggesting compatibility between the two systems without completing their connection.

The sculpture treats the Last Supper as an event rather than a single instant. From Leonardo it takes the emotional disturbance following the revelation of betrayal, allowing the apostles' individual gestures and postures to register different responses within the structure. Jesus points upward toward the Trinity, a reference to Salvador Dalí's The Sacrament of the Last Supper. The work is presented as a digital sculpture in <model-viewer>, where orbit and changing viewpoint allow the relationships among figures, nodes, cables, and voids to emerge through sustained looking.

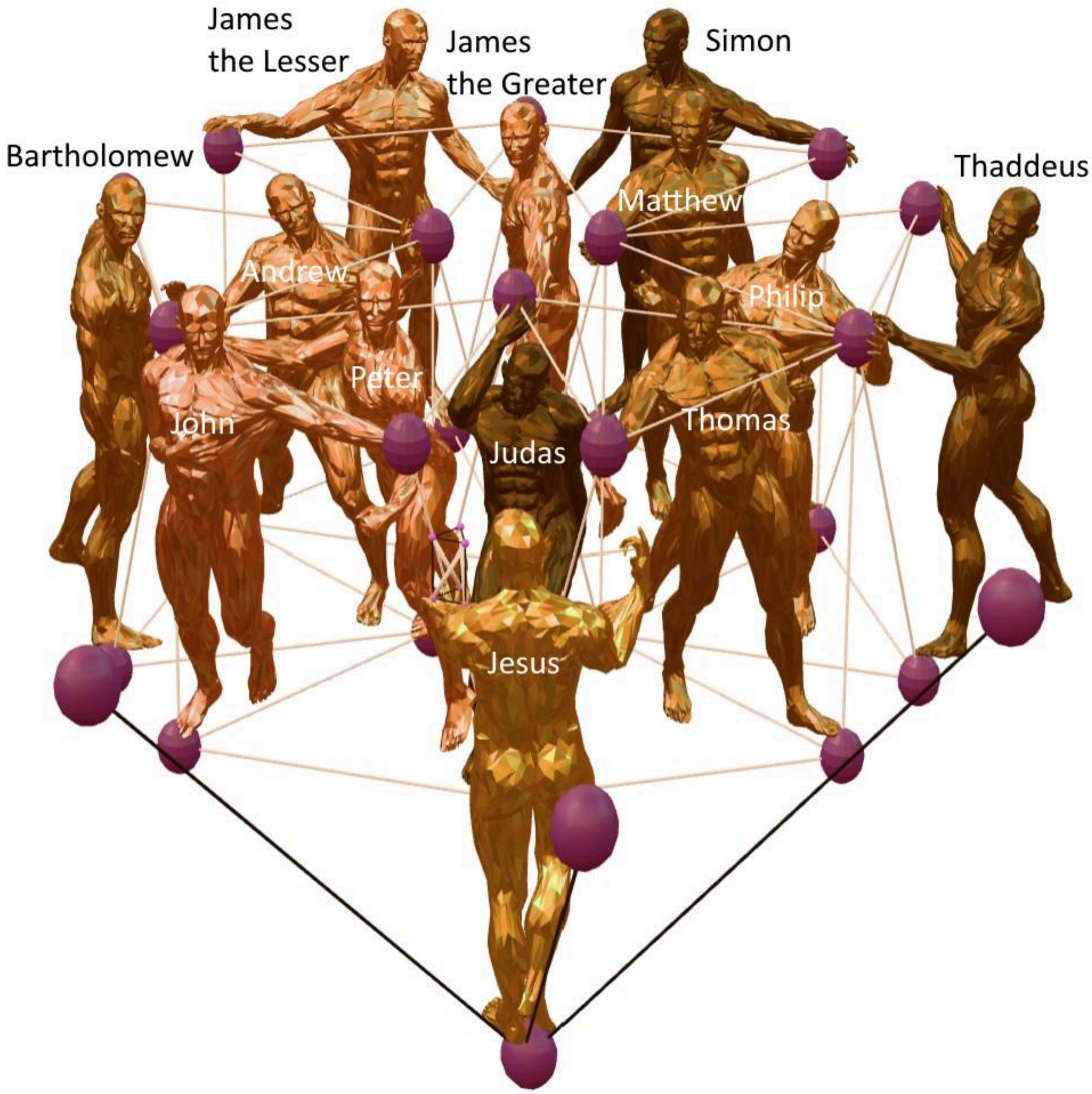
Information About the Work

The <modelviewer> Background HDR Image

The Cenacle, or the Room of the Last Supper, is a historic two-story Gothic-style hall located on Mount Zion in Jerusalem, just outside the Old City walls. Christian tradition reveres it as the site of Jesus' Last Supper and Pentecost, while the floor below is traditionally associated with the Tomb of David.
-AI overview

A good resource:
https://devotiontoourlady.com/the-cenacle-or-upper-room.html

Figure Identification



RGB Colors for the Figures

Jesus = "#DBB55D"

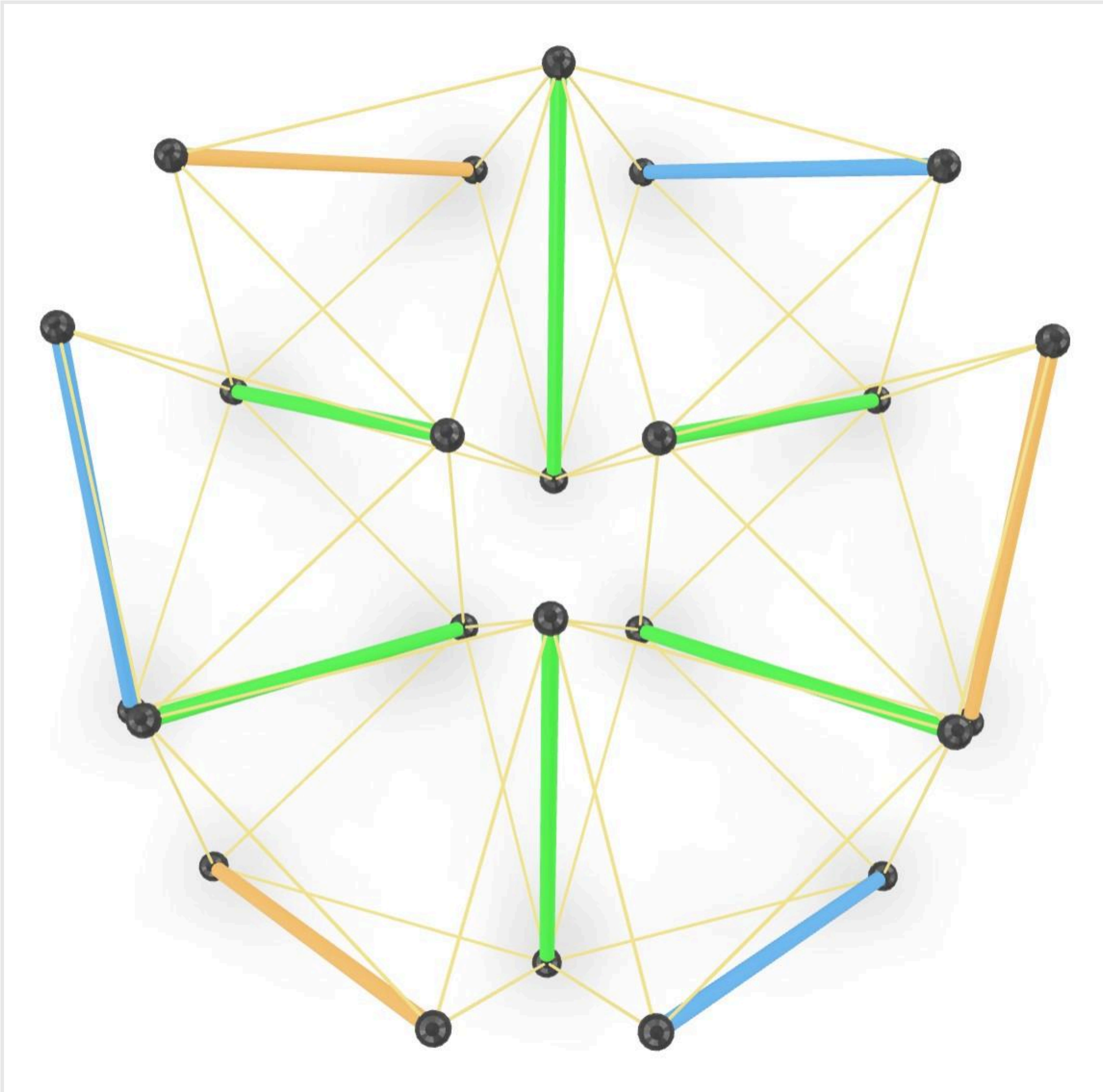
APOSTLES

Peter	"F2B287"
John	"E7AE91"

JamesTheGreater	"DCA378"
Andrew	"D19C6C"
Philip	"C69660"
Bartholomew	"BA8E55"
Thomas	"AE854C"
Matthew	"A17A43"
JamesTheLess	"936F3C"
JudeThaddeus	"846336"
Simon	"745730"
Judas	"634A2A"

Six Module Ring Color Coded

The blue struts are not shared and are part of a right-handed chirality T3 prism.
The yellow struts are not shared and are part of a left-handed chirality T3 prism.
The green struts are shared and are part of two-adjacent T3 prisms in the ring.



The TTCS Ring Solver

Theory, Design, and Development of the Six-Module, Twelve-Strut Tensegrity Ring

A technical overview for artists and art lovers

Introduction

A tensegrity prism is a small structural miracle: three rigid struts that never touch each other, floating in a web of taut cables. The simplest version, called a T3, looks like a twisted triangular drum. It is stable, light, and — because the struts are held apart purely by tension — visually paradoxical.

This document describes a body of research and development around a deceptively simple question: *what happens when you chain these prisms together?* Specifically, it traces the path to one particular object — a closed ring of six T3 prisms sharing twelve struts among them — from an initial coding scheme, through hand-building in Blender, to a validated computational solver that closes the ring to mathematical exactness and reveals structural facts nobody anticipated at the start.

The work unfolded across three research sessions, each with its own focus: a **coding scheme analysis** session that developed and refined the notation itself (culminating in the TTCS specification, draft v0.3); a **T3 ring solver** session that built and validated the numerical machinery for closing rings, documented in `TTCSRingSolverSESSION_HANDOFF.md`; and a **general solver** session that extended the machinery from rings to arbitrary prism assemblies, documented in `TTCSGeneralSolverSESSION_HANDOFF.md`. This overview synthesizes all three.

A note on the spirit of the project: this is art research conducted with engineering discipline. Every structural claim below was verified numerically — actual node positions, actual angles, actual matrix ranks — and several plausible-sounding first answers turned out to be wrong until that verification caught them. That habit became a finding in its own right.

Grammar

TTCS — the Tensegrity Coding Scheme — is a notation for describing how prism modules join together. Its central insight is a separation of two questions that are easy to blur: **topology** (which prisms share which struts) and **geometry** (where everything actually sits in space). TTCS specifies the first completely and says nothing about the second. Many different physical shapes can satisfy the same code.

The notation itself is compact. Each line of a code describes one prism module. Letters name struts that are shared between two neighboring modules; numbers count a module's open, unshared struts; commas separate the entries. A slash between two letters means both struts are shared with the same neighbor across a single flat face. A header line gives the total strut count and serves as a checksum. The six-module ring that anchors this whole project is written:

```
12
A,1,B
B,1,C
C,1,D
D,1,E
E,1,F
F,1,A
```

Read it as six modules, each sharing one strut with the module before it and one with the module after, with one open strut in between — and the letter A appearing in both the first and last lines, closing the loop. Twelve struts total: six shared, six open.

The grammar distinguishes two ways modules can join, and the difference is structural, not cosmetic. A **string** joins prisms by shared struts; each single-strut junction behaves as a hinge, free to rotate about the shared strut. A **tower** joins prisms by shared triangular bases — three connection points — which locks out rotation entirely. Strings are flexible and admit rings; towers are rigid but cannot curl. The ring above is a string. One more rule shapes everything downstream: chirality — the handedness of each prism's twist — alternates between neighbors along a string, like footprints alternating left and right.

The slash connector deserves special mention because its meaning was refined late in the work. It was originally read as "two full struts shared." The general-solver session discovered this is geometrically impossible for the T3 proportions between opposite-handed neighbors — and the resolution was a clarification of intent, not a bug fix. The slash actually expresses **SPSN** (Shared Pair-strut, Shared Node): one strut fully shared as an ordinary hinge, plus a second strut sharing only its near endpoint, which lands at one of exactly two special rotation values of that hinge. Full two-strut sharing turns out to be a special case of SPSN that only certain prism proportions permit — a genuinely deeper understanding of what the notation was pointing at all along.

General Solver

The solver is the machinery that turns a TTCS code into actual 3D geometry. It grew in stages, and the story of how it grew is inseparable from how it works.

The first attempts tried to reconstruct the ring from verbal descriptions of the hand-building procedure — overhead views, fold angles measured with a protractor. Several plausible conventions were tried and every one produced rings that closed on paper but came out tangled or unevenly spaced. The breakthrough came from abandoning description in favor of **ground truth**: the ring was rebuilt by hand in Blender with systematic naming, every node's real position was exported to JSON, and the prism geometry was mathematically fitted onto the real data. This confirmed that every module in the hand-built ring is an exact, undistorted T3 (strut length 1.0, radius 0.4, twist 150°, fit residual near one part in ten million), and it revealed that the junction-by-junction connection choices had been made individually by hand — a fact no formula would have guessed.

Ground truth also exposed the session's central bug. The original alignment function computed each junction's "roll" (how far one module rotates about the shared strut relative to its neighbor) in a way that secretly depended on where the assembly happened to sit in space. Roll values measured from the real ring wouldn't reproduce the ring when replayed from a different starting position — even though closure obviously shouldn't care where you start. The fix was a **frame-independent alignment function** that anchors each rotation to a reference point on the module itself (its centroid), verified equivariant to machine precision. With that fix, the real ring's junction rolls — previously a noisy scatter — collapsed into a tight cluster around 180°, revealing that the hand-built ring had been sitting almost exactly on a clean mathematical ideal the whole time.

The mature toolchain has several validated layers. A **parser** reads TTCS codes and performs genuine graph-based cycle detection (replacing an earlier version that matched letters textually and could be fooled). A **T3 builder** constructs the prism geometry with its verified conventions: strut k runs from bottom vertex k to top vertex $k+1$; diagonal cables run from each top vertex to the bottom vertex of the *adjacent* strut, not its own. A **ring family solver** exploits the T3's own three-fold symmetry to close uniform rings of any even module count, guaranteeing exactly planar, exactly evenly spaced results. A **rigidity analyzer** — separate machinery entirely — tests whether a solved shape can flex. A **general realizer** walks any parsed TTCS string member by member, applying the hinge mechanism at single-share junctions and the SPSN mechanism at slash junctions. And **Blender import/export scripts** move solved geometry into the studio, with the established color convention: orange struts for left-handed modules, blue for right-handed, gold cables.

Results

The six-module ring closes exactly. Starting from a uniform guess of 180° roll at every junction, the solver closes the ring with a residual around 10^{-15} — machine precision. The symmetric answer is not merely close; it is exactly right. The solved ring is perfectly planar and perfectly evenly spaced, and it matches the hand-built original to within the hand-build's own placement noise. A satisfying corollary: TTCS methodology describes shared struts as pairs of coincident struts "waiting to become one," merged in a final step. At exact closure, that merge step vanishes — the paired struts already occupy identical positions, and the merge is a no-op.

The closure rule. A string ring closes if and only if every junction is single-shared and the module count is even. The even count matters because chirality alternates: each hinge fold is partially cancelled by the opposite-handed fold at the next junction, letting out-of-plane wobble cancel while in-plane turning accumulates around the loop. An odd count leaves head and tail with matching handedness and no valid way to join. Odd counts were tested directly: zero solutions, every time.

Twelve modules is the ceiling. For these prism proportions, uniform rings close mathematically at every even count tested up to eighty — but physical validity is another matter. The clearance between each junction's open struts shrinks steadily from N=6 (a comfortable 0.36 strut-lengths) to N=12 (a razor-thin but real 0.0056), then drops to exactly zero at N=14 and stays there. Past twelve, the open struts genuinely cross each other — at a decisive 83–89° angle, not a graze. This finding required its own correction: an earlier check measured only distances between strut *endpoints* and misleadingly suggested clearance recovered past N=12; the artist caught the error by eye in a rendered close-up at N=22, spotting two struts visibly overlapping. Only a true segment-to-segment distance check revealed the sharp cutoff.

Crossing the limit makes the ring rigid. The most conceptually striking result. Every valid ring (N≤12) is, despite closing, a **mechanism** — free to flex through a continuum of shapes, because each shared strut is a true hinge. The twelve-module ring has seven independent floppy modes. But past the structural limit, the incidental strut-on-strut contacts constitute physical connections the topology never specified — and when those contacts are modeled correctly (as pin constraints tying existing strut endpoints together; a first attempt that inserted new nodes *added* freedom and was caught because the result made no physical sense), the floppy-mode count drops to exactly zero at N=14 and N=22 alike. The self-intersection that makes the ring unbuildable is precisely what makes it rigid. Flexibility and buildability, at these proportions, come as a package.

Proportions matter more than expected. Late in the work, a different valid prism proportion — struts perpendicular to the base rather than obliquely twisted — was shown to permit genuine two-full-strut sharing between opposite-handed modules, something the proportions forbid. Every hard number above (the 180° roll, the N=12 ceiling, the SPSN extremes) is a property of one point in a larger family of prism shapes, not of T3 prisms in general.

Major Concepts

A handful of ideas carry most of the weight in this work, and they travel well beyond tensegrity.

Topology versus geometry. The code says who touches whom; space decides what shape that makes. Keeping these separate is what lets a single notation describe a family of forms — and what makes questions like “does this code close?” well-posed at all.

Closure versus rigidity. A ring that closes is not a ring that holds its shape. Closure is one constraint; the hinges retain many freedoms beyond it. A closed string ring is a linkage — imagine a bicycle chain joined into a loop, closed but floppy — not a framework.

Chirality alternation. The left-right-left handedness rhythm along a string is not decoration; it is the mechanism by which rings can close at all, and the reason only even counts work.

SPSN as the general junction. One shared node, at the right hinge angle, forces a second near-coincidence — and full face-sharing is just the case where geometry lets that forcing go all the way. The notation's slash expresses an intent toward stability, not a literal topology.

Ground truth over description. Hours of careful reasoning from verbal accounts and protractor photos produced only plausible failures. One export of real coordinates resolved everything. Adjacency was *discovered* from data, never assumed.

Numeric verification as the standard of proof. The cable-wiring bug, the frame-dependence bug, the contact-model bug, the endpoint-versus-segment bug — every one produced plausible numbers from clean-looking derivations, and every one was caught by checking actual quantities against reality or against a hand-workable toy case. Symbolic re-derivation caught none of them; one bug was even re-derived into a second time under a different-looking formula. The working standard now: no geometric claim is trusted until verified numerically.

Issues

The work is mature but open at the edges. In rough priority order:

Hardcoded numbers need honest labels. The 180° roll, the SPSN extreme values, and the N=12 ceiling are all specific to one prism proportion. Code comments and docstrings currently state some of them as if general — including a retired function's claim of a chirality impossibility that the perpendicular-strut finding disproved. A documentation pass is owed.

SPSN is not yet a first-class option. Driving an ordinary junction to its SPSN roll value currently requires computing and hardcoding the value manually. A `roll_strategy="spsn"` switch is a small, well-scoped addition.

The design document is a version behind. The general realizer's design doc predates the SPSN architecture, the slash-means-intent resolution, the cable-system addition, and the proportion-generality finding. It needs a full rewrite, not a patch.

Closure solving is not yet wired into the general realizer. The realizer propagates geometry along a string but does not yet invoke the ring family's closure solve on cycles the parser flags as eligible.

Spec reconciliation is incomplete. The v0.3 spec's older fold-angle language (the 0–120° range) does not map cleanly onto the validated roll and cable-angle conventions, and the exact-closure note drafted for the spec's merge section awaits confirmation of incorporation.

Mixed prism orders remain fully open. Whether the notation survives a T3-meets-T4 junction — the spec's own first open question — is untested, and nothing in any session changes that.

Two deeper structural checks remain unrequested. The rigidity results treat cables as rigid bars, an optimistic assumption; a full proof for the rigid N≥14 rings would verify a proper self-stress state (every cable in tension, every strut in compression). And the pre-merge gap parameter — for rendering shared struts as visibly separate pairs — was lost in the solver rebuild and would need reintroduction against the corrected alignment function if that representation is wanted again.

Prepared from the TTCS specification (draft v0.3) and the session handoffs `TTCSRingSolverSESSION_HANDOFF.md` and `TTCSGeneralSolverSESSION_HANDOFF.md`.