

Interactive Design of Modular Tensegrity Characters

Damien Gauge¹ Stelian Coros² Sandro Mani¹ Bernhard Thomaszewski²

¹ETH Zürich ²Disney Research Zurich

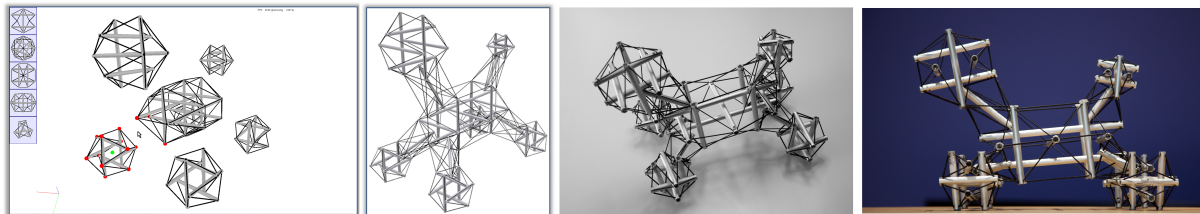


Figure 1: Our method allows users to create tensegrity characters from a library of building blocks. From left to right: snapshot from the design session, finished character, and two views of the fabricated prototype of the Puppy character.

Abstract

We present a computational design tool for creating physical characters using tensegrities — networks of rigid and elastic elements that are in static equilibrium. Whereas the task of designing general tensegrities is very difficult to automate, we show that a modular design paradigm allows users of our system to intuitively build intricate structures that approximate the shape of complex characters. The underlying concept exploited by our method is that simple tensegrities can be used as building blocks to create increasingly complex figures. To this end, we propose a dedicated optimization scheme in order to compute the lengths of elastic cables such that, when fabricated, the structure assumes the desired shape under gravity while remaining compliant and responsive to user interaction. We demonstrate the flexibility of our system by designing several types of tensegrity characters, one of which we also fabricate.

1. Introduction

Tensegrities are assemblies of strut elements carrying compressive loads and cable elements that exert tensile forces. The name *tensegrity* is a contraction of the words *tension* and *integrity*, implying that the structure is stable due to the interplay of tensile and compressive forces. In their strictest definition, all struts must be disjoint, leading to a system of floating compression elements immersed in a network of tensioned cables. Although the discussion about the origins of tensegrities remains controversial, their discovery is attributed to artist Kenneth Snelson and inventor Buckminster Fuller [SAP*01].

Thanks to their lightweight nature, high stability, and aesthetic appeal, tensegrities find application in the arts, architecture, robotics, furniture design, and many other areas. Moreover, by choosing elastic materials for the cables, one

can create compliant structures that move in response to external stimuli. Our work is motivated by the idea of using tensegrities to create tangible representations of digital characters—an emerging subfield of computer graphics that is receiving an increasing amount of attention. In particular, we are interested in designing figures that assume a desired shape under self-weight, but remain overall flexible and compliant to user interaction.

Despite their desirable properties, the use of tensegrities remains limited, arguably because their design is very difficult. Indeed, finding a tensegrity structure that approximates a given input geometry proves to be a formidable task: the strict topological constraints, high-dimensional parameter spaces and nonlinear nature of the structural forces lead to mixed continuous-discrete optimization problems which are notoriously difficult to solve.

Overview & Contributions In this work, we approach the problem of designing tensegrities in a way that side-steps the difficulties arising from the general topology problem. Using our interactive design system, the user selects building blocks from a library of template structures in order to incrementally build up tensegrity figures. Our interface provides simple tools to shape and connect individual building blocks in an intuitive way. Once the design of a tensegrity figure is complete, our optimization method computes the rest lengths for all the elastic cables, such that the structure assumes its initial shape under gravity. We evaluate our method by designing a number of complex tensegrity figures, one of which we also fabricated as a proof of concept.

The two main contributions of this paper are 1) a modular approach for designing tensegrity figures using a library of building blocks and 2) an optimization method for computing feasible tensegrity structures. The output of our method, the geometry of the struts and the rest length of the elastic cables, can be used to create physical prototypes of the tensegrity characters we design.

2. Related Work

Computational Design in Graphics Fabrication-oriented design is an emerging subfield in graphics. As one particular trend in this field, many researchers have recently started to investigate the translation of digital characters into tangible, physical representations. Bächer et al. [BBJP12] and similarly Cali et al. [CCA*12] described methods to automatically infer the geometry of mechanical joints from skinned character meshes. Skouras et al. [STC*13] presented a method that optimizes for an inhomogeneous distribution of material parameters, as well as for a sparse set of actuation points and external forces, in order to obtain physical characters with desired deformation behavior.

Another line of works in graphics has recently begun to investigate the problem of designing mechanisms that can reproduced desired motions. Zhu et al. [ZXS*12], for instance, described a motion-guided approach for synthesizing animated mechanical toys. The method of Coros and colleagues [CTN*13] allows users to create animated mechanical characters by sketching motion curves. With a similar goal, Ceylan et al. [CLM*13] use motion capture data to automatically infer the parameters of specialized mechanisms that drive the motions of mechanical automata. The method proposed by Thomaszewski et al. [TCG*14] allows for interactive creation of complex linkage structures that mimic the motions of virtual characters.

In the context of architectural design, many computational tools have addressed the problem of designing structurally-sound surfaces, made for instance, out of masonry [WSW*12, VHWP12] or rod frameworks [SFG*13]. Strut networks can also be designed specifically to minimize the cost of 3D printing material while ensuring structural sta-

bility [WY*13]. Tensegrities are a related class of structures, where rigid struts are connected to each other only through pre-tensioned elastic elements.

Tensegrities An introduction to tensegrities can be found in the textbook by Motro [Mot03]. Many different approaches to form finding for tensegrity structures have been reported in the literature. A slightly outdated summary can be found in the review article by Tibert and Pellegrino [TP03]. Existing approaches can be grouped into analytical and numerical categories, the latter consisting of methods based on evolutionary algorithms or mixed-integer programming. Analytical methods are typically based on symmetry properties and group theory [CB98].

Generally speaking, tensegrity structures are defined by their topology (connectivity of the elements and their types), their geometry and the stresses in the elements. If the topology of the structure is known, the parameters that define its geometry and the stresses can be computed numerically as described by Estrada and his colleagues [EBM06]. Their method is similar in spirit to the optimization approach we present here. However, we employ a gradient-based method, and we iterate between computing the geometry of the tensegrities and the stresses acting along the elastic elements.

If the topology of a tensegrity is not known in advance, the problem of designing a structure that approximates an input target shape is difficult. Evolutionary algorithms for discovering irregular tensegrity structures can be employed [PLC05, RVCL09], but there are no guarantees that the quality of the results is satisfactory. The method described by Ehara and Kanno [EK10] employs a sequence of mixed-integer programs to compute a tensegrity's topology, geometry, and stresses, following the ground structure method that is widely used in optimization of discrete structures. We take a different approach to creating the topology of the tensegrity characters. Our system allows users to interactively design the figures by instantiating different types of elementary tensegrities, which are then automatically linked to each other through new rigid struts and elastic elements.

The method by Tachi [Tac12] converts an arbitrary polygonal mesh into a stable tensegrity structure approximating this shape. While this work shows impressive results in simulation, fabricating such complex irregular tensegrities is a formidable task, requiring careful planning of the assembly order as stability is only obtained once the last strut is inserted. Our modular design approach makes fabrication much easier, allowing building blocks to be assembled in isolation.

Shape Optimization The problem of computing the rest configuration of elastic structures to control their deformation behavior has been investigated both in the context of animation and fabrication-oriented design. Twigg and Kacic-

Alesic [TKA11] compute the rest lengths of springs for clothing simulation in order to obtain a desired drape under gravity. With a similar goal but in the context of hair simulation, Derouet-Jourdan and colleagues described methods to compute rest shapes for elastic rods subject to gravity [DJBDT10] and frictional contact [DJBDT13]. The example-based method by Martin et al. [MTGG11] modulates rest shapes of volumetric objects in order to guide them towards a desirable space of deformations. In the context of fabrication-oriented design, Skouras et al. [STBG12] describe a method to compute the rest shape of rubber balloons that, when inflated to a target pressure, assume a desired shape. Similar in spirit to these works, our method optimizes for the rest lengths of the elastic elements such that the tensegrity figures designed in our system can support their own weight. Together with the geometry of the rigid struts, this provides sufficient information to fabricate the characters.

3. Computational Modeling of Tensegrity Structures

A tensegrity character is defined through a set of nodes \mathcal{N} , a set of elements \mathcal{E} connecting the nodes as well as a vector σ holding the stresses (or force magnitudes) carried by the elements. The position of the n nodes are denoted by \mathbf{x}_i , with \mathbf{x} referring to the vector holding all nodal positions. Each structural element $e_{ij} \in \mathcal{E}$ connects two nodes, \mathbf{x}_i and \mathbf{x}_j , and we distinguish between strut and cable elements. In any given configuration of the tensegrity, each non-redundant element e_{ij} will exhibit a non-zero scalar stress $\sigma_{ij} \neq 0$. Struts are required to exhibit compressive stresses ($\sigma_{ij} < 0$) whereas cables can only support tensile stress ($\sigma_{ij} > 0$).

The elastic energy in a given cable element e_{ij} can be modeled as

$$E_{ij} = \frac{1}{2}k(\varepsilon_{ij})\varepsilon_{ij}^2L_{ij} \quad \text{with} \quad \varepsilon_{ij} = \frac{l_{ij} - L_{ij}}{L_{ij}}, \quad (1)$$

where ε_{ij} is the deformation of the cable, $k(\varepsilon_{ij})$ is a deformation-dependent stiffness coefficient, $l_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the current length, and L_{ij} the undeformed lengths of the cable, respectively. As usual, nodal forces are obtained as the negative gradient of this energy with respect to \mathbf{x}_i and \mathbf{x}_j . This generic formulation allows for incorporating measured material data and will be used for dynamic simulation. For the purpose of tensegrity design, however, we will resort to a simpler force expression in terms of force magnitudes σ_{ij} ,

$$\mathbf{f}_{ij} = \sigma_{ij} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}. \quad (2)$$

For notational convenience, we use the convention $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$ and $\sigma_{ij} = \sigma_{ji}$. The net force acting on a given node i of the tensegrity structure is obtained by summing up all force contributions of incident elements as well as gravitational forces

\mathbf{f}_g and possibly contact forces \mathbf{f}_c as

$$\mathbf{f}_i = \sum_j \mathbf{f}_{ij} + \mathbf{f}_g + \mathbf{f}_c. \quad (3)$$

In order for a tensegrity structure to be in equilibrium, the net forces have to sum to zero at every node, i.e., $\mathbf{f}_i = 0 \forall i$. In addition, the sign of the stresses has to match the element type, i.e., positive for cables and negative for struts. These conditions also apply for nodes that are in contact with the ground, but for simplicity, we assume that contact nodes are known in advance, that they remain fixed, and they are treated as bilateral constraints. We can thus assume that the contact forces \mathbf{f}_c cancel out all other forces such that we do not need to consider contact nodes. For all non-contact nodes, we combine the above conditions into a Quadratic Program (QP) as

$$\arg \min_{\sigma} \|\mathbf{f}(\mathbf{x}, \sigma)\|^2 \quad \text{s.t.} \quad \sigma_{ij} \begin{cases} > 0 & e_{ij} \text{ is cable,} \\ < 0 & e_{ij} \text{ is strut.} \end{cases} \quad (4)$$

Feasibility While this approach is viable for many shapes, not all geometries can be realized as tensegrities. For a given node, let \mathbf{e}^s be the edge vector corresponding to the single incident strut element and let \mathbf{e}_j^c denote edge vectors corresponding to the cable elements. If there is no set of non-negative weights λ_j such that $\mathbf{e}^s = \sum \lambda_j \mathbf{e}_j^c$, then there is no combination of admissible stresses that lead to force equilibrium. For such geometries, the solution of (4) will have a non-zero objective, implying force imbalance. A natural way of dealing with such cases is to seek the smallest change in geometry that leads to a feasible configuration. This can again be formulated as a constraint optimization problem,

$$\arg \min_{\mathbf{x}, \sigma} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \quad \text{s.t.} \quad \mathbf{f}_i(\mathbf{x}, \sigma) = 0 \forall i, \quad (5)$$

where the sign constraints on the stresses are omitted from the equation for simplicity. The nature of the force constraints leads to a nonlinear and nonconvex optimization problem, which is significantly more challenging to solve than the QP (4). In order to render this problem tractable, we take an alternating optimization approach and optimize (4) for position and stress variables separately.

Alternating Optimization Instead of directly solving the nonlinear program (5), we take an iterative approach in order to minimize the objective of (4) for both position and stress variables while satisfying the sign constraints on the stresses.

Algorithm 1 describes our alternating optimization method. In each iteration, we first solve the QP (4) for the stress variables while holding the current positions fixed. Since this step is not guaranteed to lead to a feasible tensegrity with zero forces at each node, we subsequently reduce the norm of the residual forces with a few steps of gradient descent on the position variables. We note that it is also possible to use Newton's method in this second step in order to directly jump to a local minimum of the objective function. However,

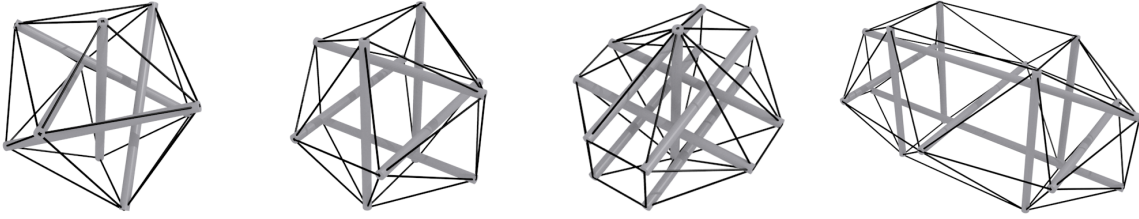


Figure 2: An overview of some building blocks provided by our system (from left to right): quadruplex, icosahedron, diamond, and double icosahedron.

Algorithm 1 Form Finding with Alternating Optimization

```

1: while  $\|\Delta\mathbf{x}\| > \varepsilon_{cvg}$  do
2:    $\sigma = \text{solveQP}(\mathbf{x}, \sigma);$  //Eq. (4)
3:    $\gamma_{acc} = \gamma, i = 0;$ 
4:   while  $\gamma_{acc} > 0$  and  $i < i_{max}$  do
5:      $\Delta\mathbf{x} = \text{computeDescentDirection}(\mathbf{x});$  //Eq. (6)
6:      $\alpha = \text{lineSearch}(\mathbf{x}, \Delta\mathbf{x}, \gamma_{acc});$ 
7:      $\mathbf{x} += \alpha\Delta\mathbf{x};$ 
8:      $\gamma_{acc} -= \alpha, i++;$ 
9:   end while
10: end while
  
```

this local minimum can be quite far away from the current configuration. Moreover, our experiments showed that even small changes in geometry are often sufficient to transform an infeasible configuration into a feasible tensegrity. In order to find a feasible configuration that is close to the current configuration, we proceed along the current gradient of the objective function and impose a threshold γ on the maximum distance traveled. The corresponding displacement is computed as

$$\Delta\mathbf{x} = \gamma \frac{(\nabla_{\mathbf{x}}\mathbf{f})'\mathbf{f}}{\|(\nabla_{\mathbf{x}}\mathbf{f})'\mathbf{f}\|}. \quad (6)$$

With $\Delta\mathbf{x}$ determined, we then perform line search in order to ensure that the objective is decreased. The process terminates once the norm of $\Delta\mathbf{x}$ falls below a given threshold ε_{cvg} . Since both optimization steps monotonically decrease the objective function of (4), this alternating optimization scheme is guaranteed to converge to a local minimum. While there is no guarantee that the objective function vanishes at this minimum, we did not encounter such a case in practice.

Stiffness Scaling Using the alternating optimization method described above, we can compute the geometry and a corresponding set of forces that lead to a stable tensegrity character. Without further assumptions, however, the force magnitudes are not unique. This can be easily observed by temporarily omitting gravity: if (3) holds for every node, then it will also hold when scaling all forces by a constant factor. We can exploit this null-space in order to determine the overall stiffness of the tensegrity structure. In practice,

we add a weak quadratic regularizer to (4) that asks for small force magnitudes. After a solution has been found, we allow the user to specify a desired deformation that the elastic cables should exhibit at equilibrium. We use these deformations to compute corresponding target stresses for each cable. We then formulate another QP, where the quadratic objective asks that the stress in each cable is as close as possible to its target value, subject to hard constraints on force equilibrium and the sign of the stresses in struts and cables. Since we solve this QP once a stable tensegrity structure is found, we are sure that the force equilibrium constraint can be satisfied as a hard constraint.

Simulation Once the design of a tensegrity character is complete, the user can evaluate its deformation behavior in an interactive simulation session. For this purpose, we extend the force model described in (3) to inertia forces, using a simple mass lumping scheme that distributes mass for each strut and cable evenly to their nodes. User interaction is incorporated through spring-like forces applied to the nodes of the structure. We solve the resulting equations of motion in time using the implicit Euler scheme,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}_n + h^2\mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_{n+1}), \quad (7)$$

where \mathbf{M} denotes the mass matrix, h is the step size, and the subscripts indicate the time step to which the quantities pertain. We solve this system of nonlinear equations using Newton's method.

4. Modular Tensegrity Design

Building Blocks Our modular approach to tensegrity design greatly simplifies the form finding problem. The user builds up characters incrementally by selecting building blocks from a predefined library. A few example of such building blocks are shown in Fig. 2. Our interface allows the user to transform the building blocks as desired, either through global scaling or by displacing nodes individually. The template library we use provides several well-known types of tensegrities, and can be easily extended using available data sources such as the catalogue published by Connelly and Black [CB98].

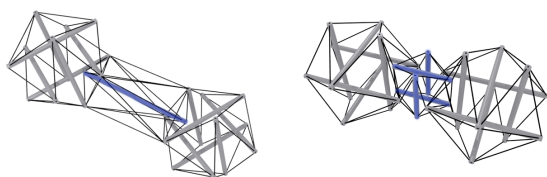


Figure 3: Different ways of connection two building blocks: a single face connection (left) and a double face connection (right). Connecting struts are shown in blue.

Connections Pairs of building blocks are linked to each other through connections of predefined topologies. We currently support two types of connections, *face links* and *edge links* (see Fig. 3). For a face link, 12 cables and a single strut are inserted such as to connect a triangular face on one building block to one on the other module. The ends of the struts are attached to the corresponding triangles through three cables on each side, whereas the remaining six cables provide the required tension between the building blocks. It should be noted that in order for this connection to be stable, the strut has to pass through the connecting faces on each building block. The optimization method we describe automatically computes the geometry of the connections in order to lead to stable structures.

An edge link removes a cable on both building blocks, adding four struts and 38 cables. As illustrated in Fig. 4, this connection can be interpreted as inserting an icosahedron in between the two building blocks, first *merging* (Fig. 4, left) and subsequently removing the two overlapping struts and cables (Fig. 4, right).

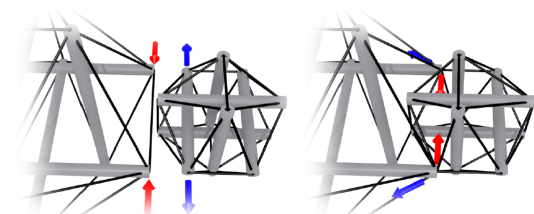


Figure 4: Adding an edge link can be interpreted as inserting an icosahedron. The red/blue arrows indicate tensile/compressive forces before (left) and after the operation (right).

With two connection types available, the question is when to use which type. One important factor is the relative orientation of the building blocks that should be connected. Another one is the desired range of motion at the connection. The face link exhibits a more or less isotropic resistance to deformation, making it best suited for spherical joints. The edge link exhibits lower resistance for rotations around the vertical strut direction, which is similar to hinge joints. In

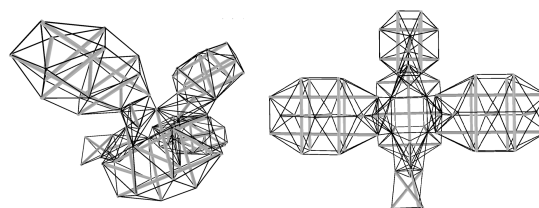


Figure 5: Two views of our Duck model, which was constructed from 5 building blocks, 4 face links and a total of 41 struts and 253 cables.

our interface, we let the user decide which type of link is best suited for a given connection.

5. Experiments and Results

We used our design system to create a number of tensegrity characters, three of which we briefly describe. The *Duck* character (Fig. 5) was constructed from 5 building blocks, resulting in a network of 41 struts and 253 cables. The *Scorpion* figure (Fig. 6) is our most complex result: 12 building blocks and 11 face links led to a tensegrity structure with 83 struts and 523 cables. The supplementary video shows the full process of creating the *Puppy* character (Fig. 1), including the physical prototype we built.

Stiffness Design Our system takes a single shape as input and the cable stresses are computed such that the desired shape is assumed under gravity. If elastic cables are used for fabrication, the shape of the figure will likely not be preserved as the tensegrity is rotated and gravity acts along a different axis than during the design process. The stability of the shape can be improved by using stiffer cables but, unsurprisingly, this will also have an effect on how compliant the character is to user interaction. With our interface, the user can experiment with different cable materials and investigate the resulting deformation behavior in a dynamic simulation. The accompanying video includes an editing session in which we tested this stiffness design mode on the *Scorpion* and on the *Duck* characters. We currently only provide visual feedback of the interactions to the user, but the experience could be further improved by using a haptic input device.

Performance For all the examples presented in this paper, our system runs at interactive rates for both the design and simulation stages. The rest length optimization takes only a few seconds to complete, and the characters we describe in this paper took between 5 and 10 minutes to design.

Optimization All of the examples that we designed, including those discussed in this section, exhibit significant variations in the deformation of the cables. For the puppy character, for instance, the deformations computed by our method

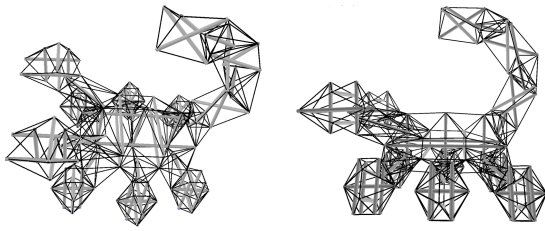


Figure 6: Two views of the Scorpion character, consisting of 12 building blocks and 11 face links with a total of 83 struts and 523 cables.

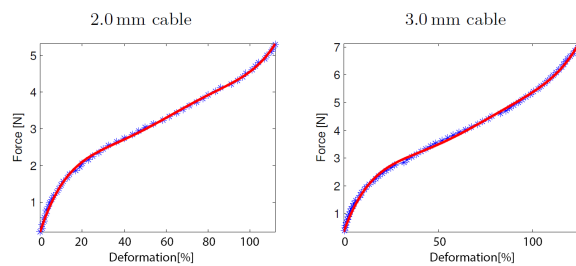


Figure 7: Force-deformation plots for different cable diameters. The blue crosses indicate measurements and the red curves are corresponding 5th-order polynomial fits.

range between 7% and 98%. For comparison, we also experimented with a simple approach that merely sets the rest lengths of the cables to a constant fraction of their deformed length. However, we were not able to find such a scaling factor that would allow the characters to maintain their shape under gravity.

5.1. Fabrication

Cable Lengths In order to manufacture the tensegrity figures designed with our method, we first have to determine the rest lengths for the structural elements. Since the struts are much stiffer than the cables, their deformations are negligible and their rest lengths can be obtained directly from the geometry. However, unlike most other applications of tensegrities, we use elastic cables in order to build compliant characters that are responsive to user interaction. Consequently, the cables can exhibit significant deformations and we have to compute their rest lengths such that the desired internal forces are exerted. For this purpose, we first have to determine the force-deformation behavior of the cables that are to be used for fabrication. We experimented with a particular type of cable that features an elastic core and nylon coating. As seen from Fig. 7, the force-deformation relation is highly non-linear, but it can be well-approximated by a 5th order polynomial curve. With this material model established, and given the deformed length and the force exerted

by each cable, we can compute the rest lengths of the elastic elements in a straightforward manner.

Materials We experimented with different ways of fabricating the tensegrity structures designed with our system. We first attempted to 3D print the rigid struts, whose geometry was procedurally generated and included cable pathways at the endpoints. When using Vero-type materials, the struts underwent significant plastic deformations under the tension of the cables, as seen in Fig 8. In addition, the relatively brittle materials currently used for 3D printing required bulky designs for the cable connections. We therefore chose to use hollow aluminum tubes to create the rigid struts, as illustrated in Figs 9 and 10.

Assembly One important advantage of our modular design system is that the fabrication process is greatly simplified. In a general setting, a tensegrity structure is stable only when all the cables and struts are connected to each other. As the structure of a tensegrity becomes more complex, assembling it therefore becomes increasingly more challenging. With our approach, intermediate structures (i.e. individual building blocks or connections of building blocks) are stable and can be manufactured independently using the same sequence of steps that were used to design the digital models.



Figure 8: After two weeks, the 3D printed struts have deformed significantly under the tension of the cables.

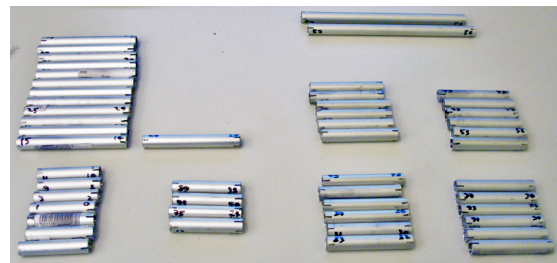


Figure 9: The 51 struts used for manufacturing the Puppy character.

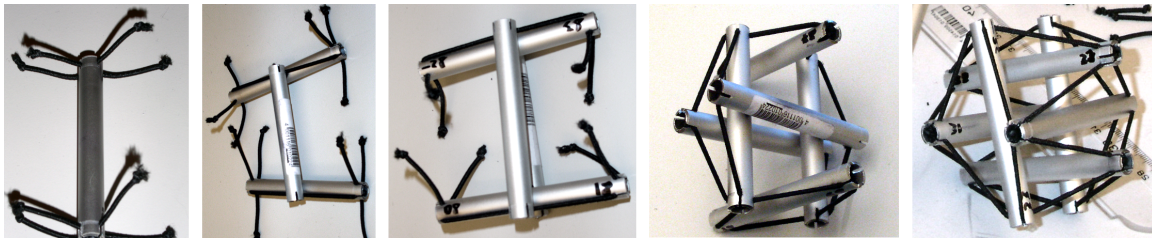


Figure 10: The order of assembly used for manufacturing the icosahedron building block.

6. Conclusion, Limitations and Future Work

This paper describes a computational design system that allows casual users to interactively create characters represented as compliant tensegrity structures. We employ a modular system, where elementary types of tensegrities are used as building blocks in designing complex figures that approximate the shape of digital characters. We describe two strategies that users can choose from to connect the individual building blocks instantiated by our framework. These two connection types exhibit different mechanical properties and anisotropic resistance to deformation. While we currently let the user decide which connections to use, it would be interesting to automatically suggest the connection type that best fits a desired deformation behavior or range of motion. Automatically determining other types of connections represents another avenue for future work.

While our method produces tensegrity structures that are in static equilibrium, there is no guarantee that the equilibrium configurations will always be stable, i.e., robust against perturbations. While unstable solutions are easy to detect and report to the user, avoiding them altogether is challenging [ZO07]. Nevertheless, we did not encounter such configurations during our experiments, which is likely due to the fact that all of our building blocks are stable tensegrities.

Our method allows users to design tensegrity shapes, but their network structure also provides a natural means for motion control: by varying the lengths of struts or cables, different deformed shapes can be obtained. This observation is already being explored to create an interesting class of robots [PVCL06, BCI*14], and in the future, we would like to investigate the problem of computing minimal sets of actuation elements that are able to generate a desired range of motions for our tensegrity characters.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

- [BBJP12] BÄCHER M., BICKEL B., JAMES D. L., PFISTER H.: Fabricating articulated characters from skinned meshes. In *Proc. of ACM SIGGRAPH '12* (2012). 2
- [BCI*14] BRUCE J., CALUWAERTS K., ISCEN A., SABELHAUS A. P., SUNSPIRAL V.: Design and evolution of a modular tensegrity robot platform. In *IEEE International Conference on Robotics and Automation (ICRA)* (2014). 7
- [CB98] CONNELLY R., BLACK A.: Mathematics and tensegrity. *American Scientist* 86 (March-April 1998), 142–151. 2, 4
- [CCA*12] CALÌ J., CALIAN D., AMATI C., KLEINBERGER R., STEED A., KAUTZ J., WEYRICH T.: 3D-printing of non-assembly, articulated models. In *Proc. of ACM SIGGRAPH Asia '12* (2012). 2
- [CLM*13] CEYLAN D., LI W., MITRA N. J., AGRAWALA M., PAULY M.: Designing and fabricating mechanical automata from mocap sequences. *ACM Transactions on Graphics* 32, 6 (2013). 2
- [CTN*13] COROS S., THOMASZEWSKI B., NORIS G., SUEDA S., FORBERG M., SUMNER R. W., MATUSIK W., BICKEL B.: Computational design of mechanical characters. *ACM Trans. Graph.* 32, 4 (2013). 2
- [DJBDDT13] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., DAVIET G., THOLLOT J.: Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph.* 32, 6 (2013). 3
- [DJBDDT10] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Stable inverse dynamic curves. In *ACM SIGGRAPH Asia 2010 Papers* (2010), SIGGRAPH ASIA '10. 3
- [EBM06] ESTRADA G., BUNGARTZ H.-J., MOHRDIECK C.: Numerical form-finding of tensegrity structures. *International Journal of Solids and Structures* 43, 22-23 (2006), 6855–6868. 2
- [EK10] EHARA S., KANNO Y.: Topology design of tensegrity structures via mixed integer programming. *International Journal of Solids and Structures* 47, 5 (2010), 571–579. 2
- [Mot03] MOTRO R.: *Tensegrity: structural systems for the future*. Butterworth-Heinemann, 2003. 2
- [MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Trans. Graph.* 30, 4 (2011). 3
- [PLC05] PAUL C., LIPSON H., CUEVAS F. J. V.: Evolutionary form-finding of tensegrity structures. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05)* (2005). 2
- [PVCL06] PAUL C., VALERO-CUEVAS F., LIPSON H.: Design and control of tensegrity robots for locomotion. *Robotics, IEEE Transactions on* 22, 5 (2006), 944–957. 7

- [RVCL09] RIEFFEL J., VALERO-CUEVAS F., LIPSON H.: Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures* 87, 5-6 (2009), 368–379. 2
- [SAP*01] SKELTON R., ADHIKARI R., PINAUD J.-P., CHAN W., HELTON J.: An introduction to the mechanics of tensegrity structures. In *Proceedings of the 40th IEEE Conference on Decision and Control* (2001). 1
- [SFG*13] SONG P., FU C.-W., GOSWAMI P., ZHENG J., MITRA N. J., COHEN-OR D.: Reciprocal frame structures made easy. *ACM Trans. Graph.* 32, 4 (2013). 2
- [STBG12] SKOURAS M., THOMASZEWSKI B., BICKEL B., GROSS M.: Computational design of rubber balloons. In *Proc. of Eurographics '12* (2012). 3
- [STC*13] SKOURAS M., THOMASZEWSKI B., COROS S., BICKEL B., GROSS M.: Computational design of actuated deformable characters. *ACM Trans. Graph.* 32, 4 (2013). 2
- [Tac12] TACHI T.: Interactive freeform design of tensegrity. In *Advances in Architectural Geometry*. Springer Vienna, 2012, pp. 259–268. 2
- [TCG*14] THOMASZEWSKI B., COROS S., GAUGE D., MEGARO V., GRINSPUN E., GROSS M.: Computational design of linkage-based characters. In *Proc. of ACM SIGGRAPH '14* (2014). 2
- [TKA11] TWIGG C. D., KAČIĆ-ALESIĆ Z.: Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), SCA '11, pp. 225–236. 3
- [TP03] TIBERT A., PELLEGRINO S.: Review of form-finding methods for tensegrity structures. *International Journal of Space Structures* 18, 4 (2003), 209–223. 2
- [VHWP12] VOUGA E., HÖBINGER M., WALLNER J., POTTMANN H.: Design of self-supporting surfaces. *ACM Trans. Graph.* 31, 4 (2012). 2
- [WSW*12] WHITING E., SHIN H., WANG R., OCHSENDORF J., DURAND F.: Structural optimization of 3d masonry buildings. *ACM Trans. Graph.* 31, 6 (2012). 2
- [WWY*13] WANG W., WANG T. Y., YANG Z., LIU L., TONG X., TONG W., DENG J., CHEN F., LIU X.: Cost-effective printing of 3d objects with skin-frame structures. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 32, 5 (2013). 2
- [ZO07] ZHANG J., OHSAKI M.: Stability conditions for tensegrity structures. *International Journal of Solids and Structures* 44, 11–12 (2007), 3875 – 3886. 7
- [ZXS*12] ZHU L., XU W., SNYDER J., LIU Y., WANG G., GUO B.: Motion-guided mechanical toy modeling. *ACM Trans. Graph.* 31, 6 (2012). 2