# Architectural Origami

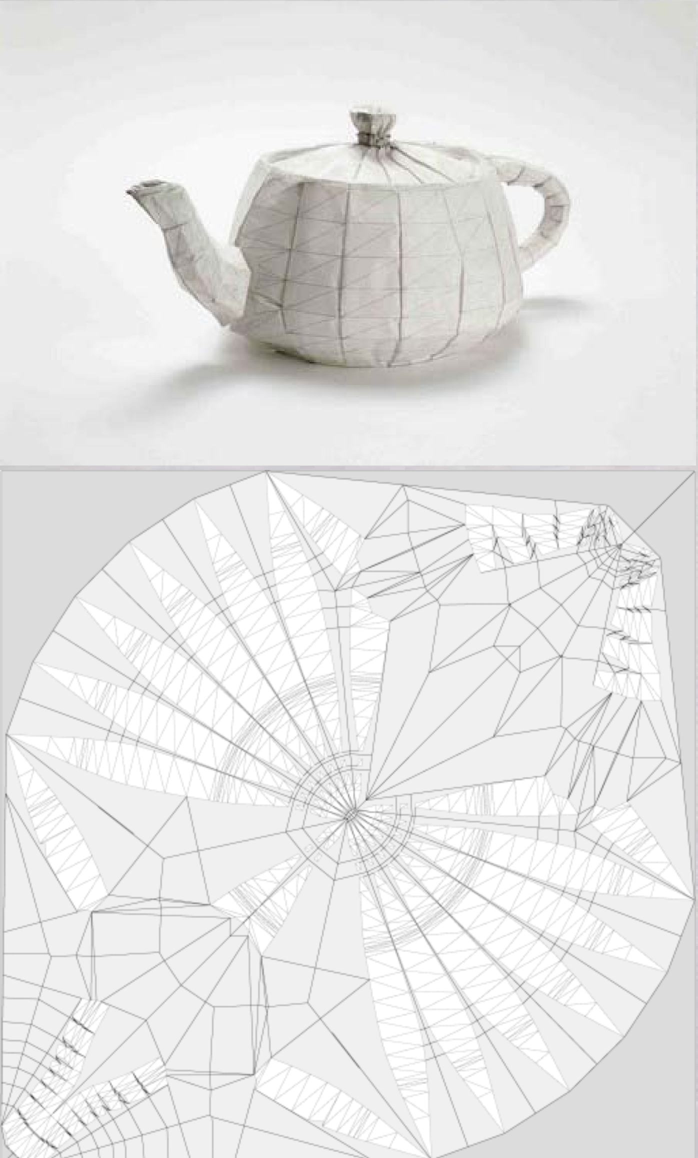
## Architectural Form Design Systems based on Computational Origami

Tomohiro Tachi
Graduate School of Arts and Sciences, The University of Tokyo
JST PRESTO
tachi@idea.c.u-tokyo.ac.jp

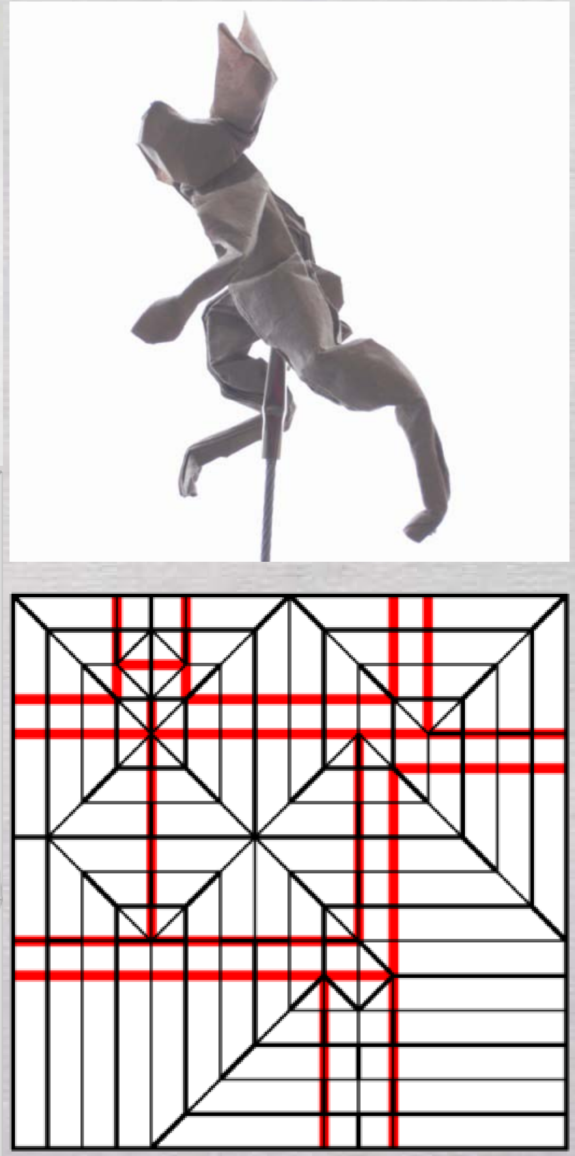
# 0

# Introduction

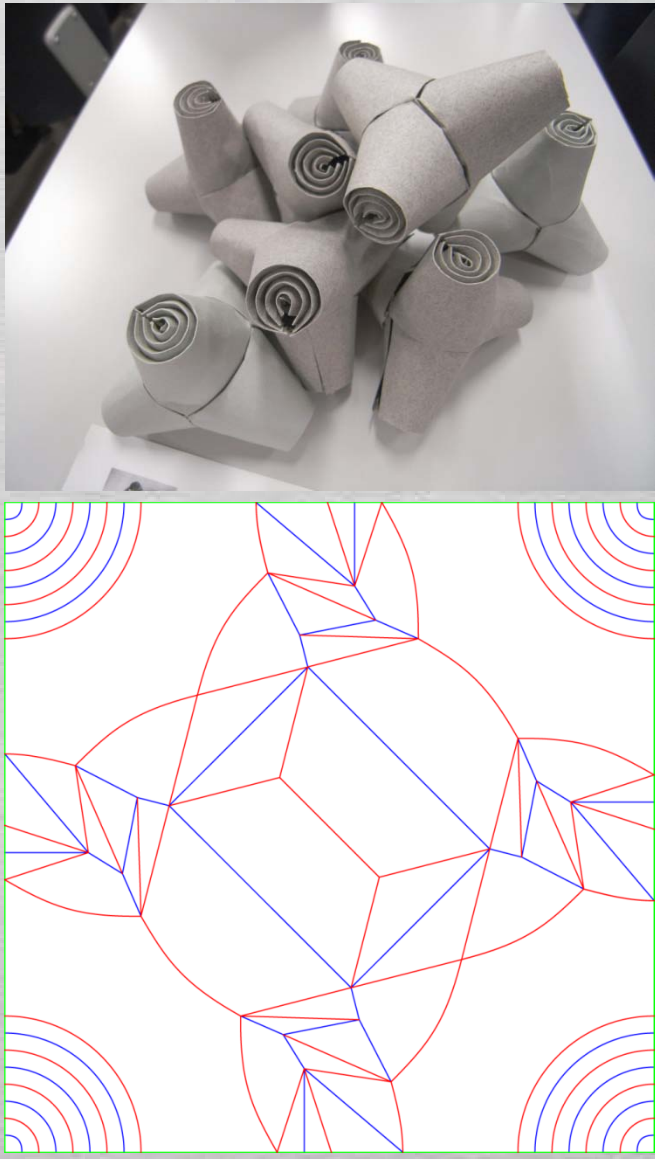
# Background 1: Origami



Origami Teapot 2007
Tomohiro Tachi



Running Hare 2008
Tomohiro Tachi



Tetrapod 2009
Tomohiro Tachi

# Background 2: Applied Origami

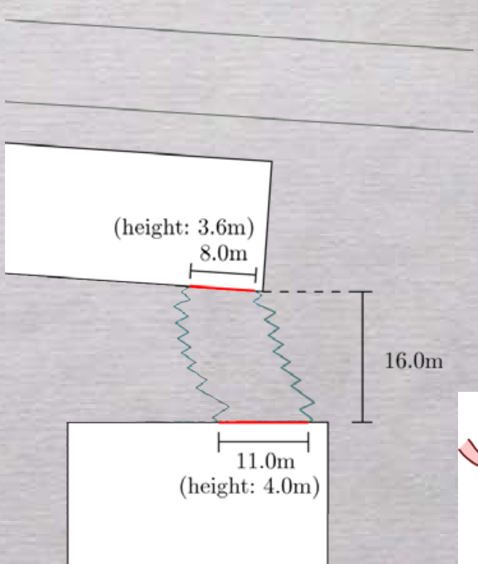
- Static:
  - Manufacturing
    - Forming a sheet
    - No Cut / No Stretch
    - No assembly
  - Structural Stiffness
- Dynamic:
  - Deployable structure
    - Mechanism
    - Packaging
  - Elastic Plastic Property
    - Textured Material
    - Energy Absorption
- Continuous surface

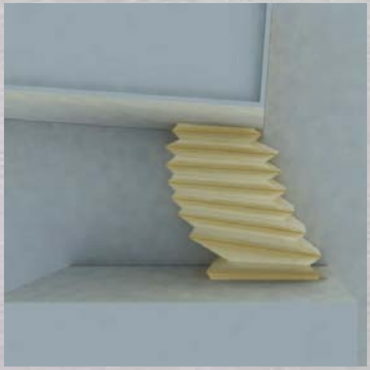
Potentially useful for
- Adaptive Environment
- Context Customized Design
- Personal Design
- Fabrication Oriented Design

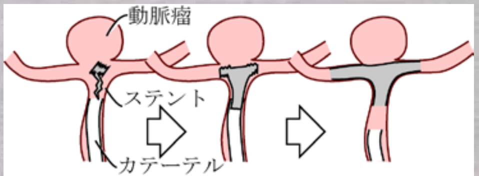
Table
(T. Tachi and D. Koschitz)

Dome (Ron Resch)

Deployable Solar panels
(K. Miura)





# Architectural Origami

- Origami Architecture

  Direct application of Origami for Design

  – Design is highly restricted by the symmetry of the original pattern

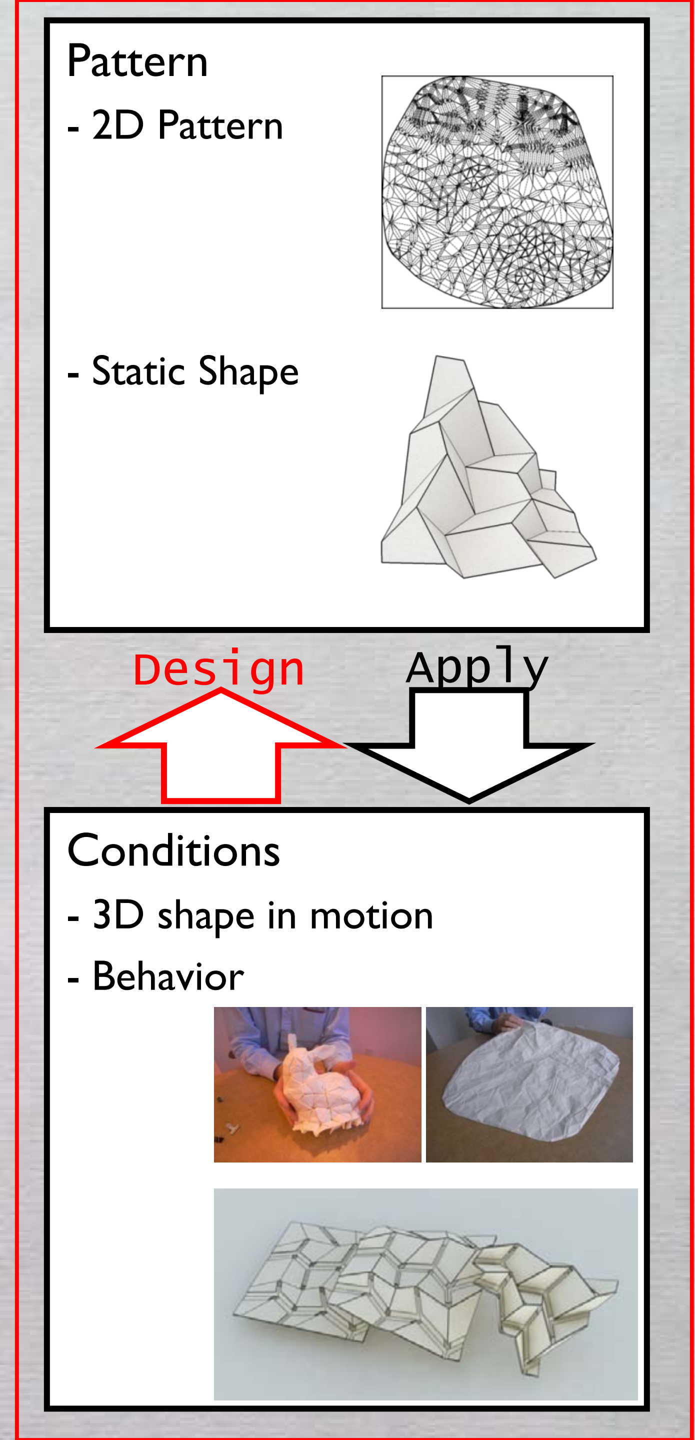
  – Freeform design results in losing important property (origami-inspired design)

- **Architectural Origami**

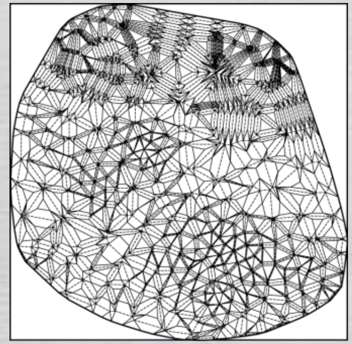
  **Origami theory for Design**

  – **Extract characteristics of origami**

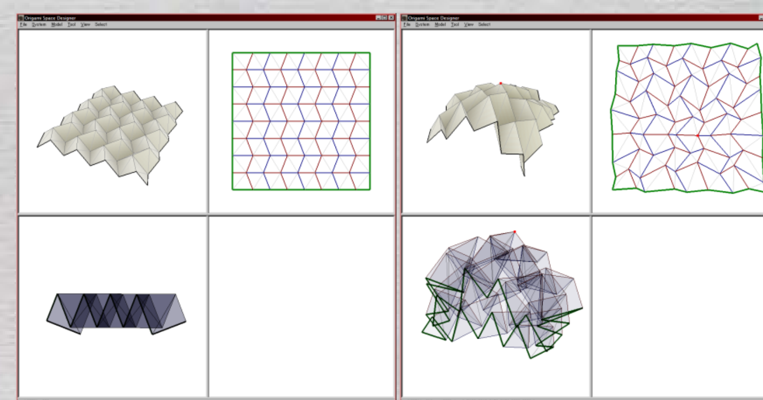
  – **Obtain solution space of forms from the required condition and design context**



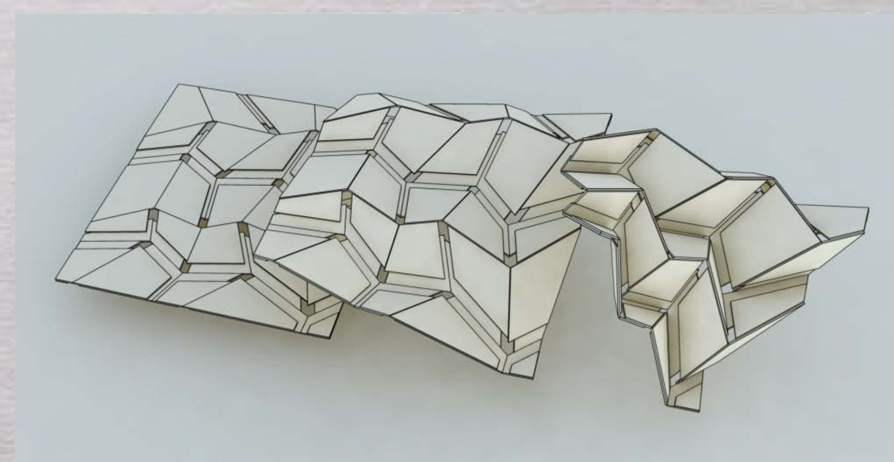
# Outline



1. Origamizer
   - tucking molecules
   - layout algorithm



2. Freeform Origami
   - constraints of origami
   - perturbation based calculation
   - mesh modification
3. Rigid Origami
   - simulation
   - design by triangular mesh
   - design by quad mesh
   - non-disk?



# 1

# Origamizer

Related Papers:

•Demaine, E. and Tachi, T. “Origamizer: A Practical Algorithm for Folding Any Polyhedron,” work in progress.

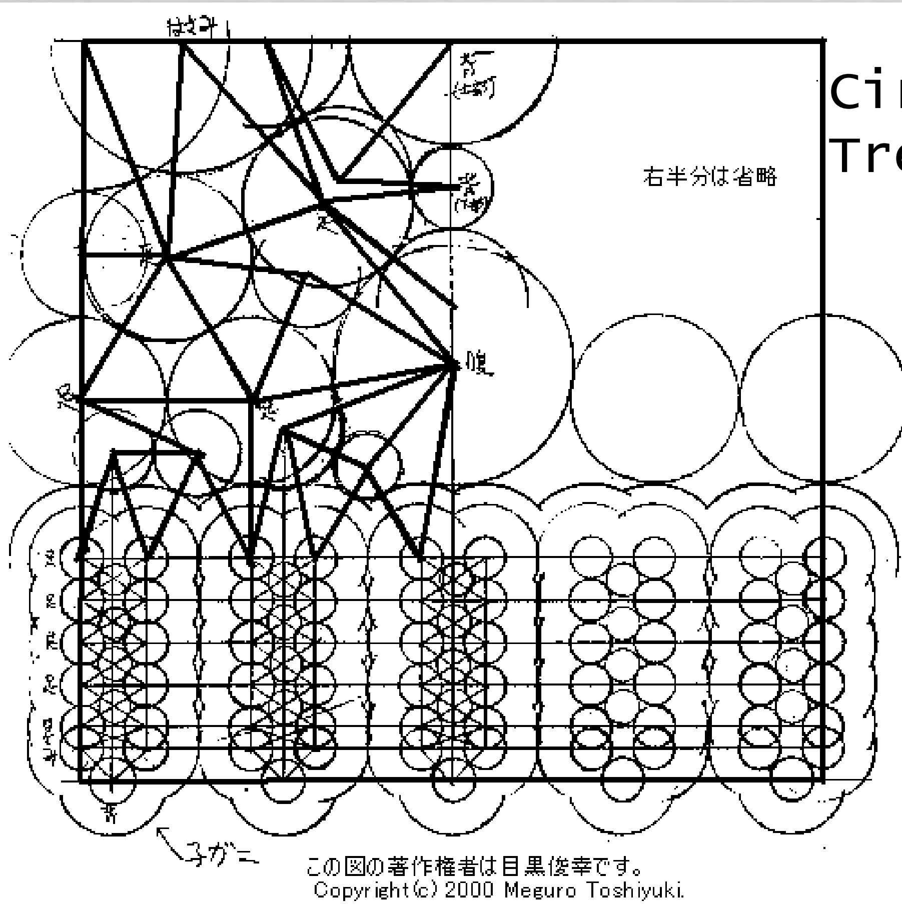
•Tachi, T.,“Origamizing polyhedral surfaces,” IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 2, 2010.

•Tachi, T., “Origamizing 3d surface by symmetry constraints,” August 2007. ACM SIGGRAPH 2007 Posters.

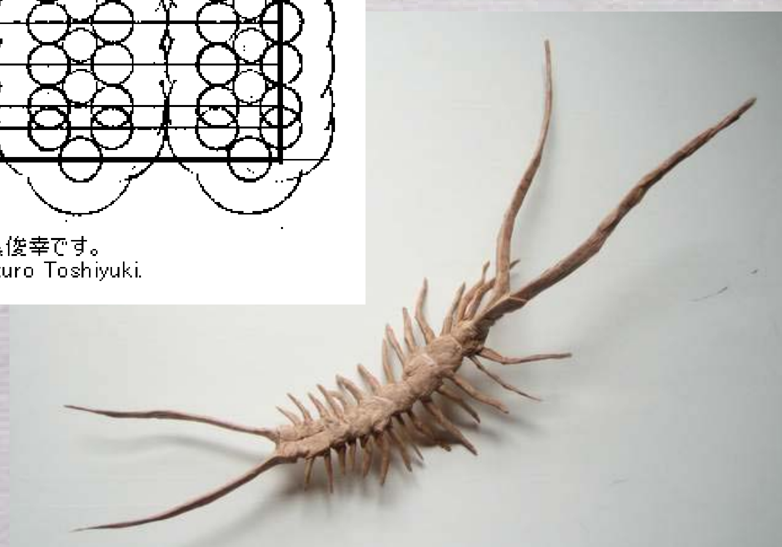
•Tachi, T., “3D Origami Design based on Tucking Molecule,” in Origami4: A K Peters Ltd., pp.  259-272, 2009.

# Existing Origami Design Method by Circle Packing

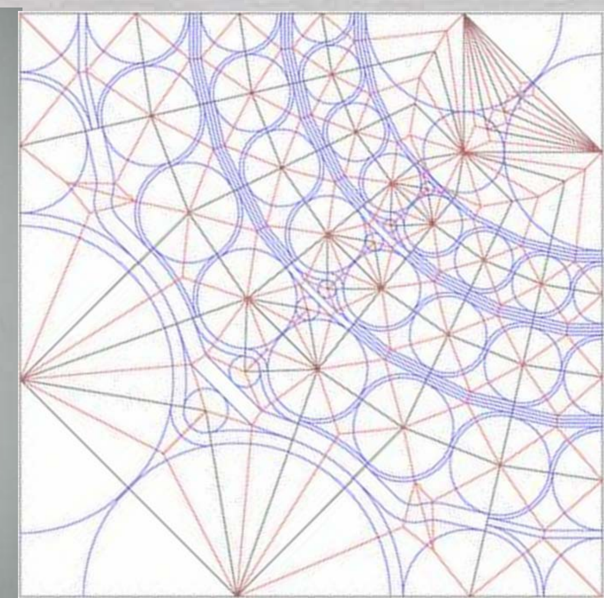
Circle River Method [Meguro 1992]
Tree Method [Lang 1994]



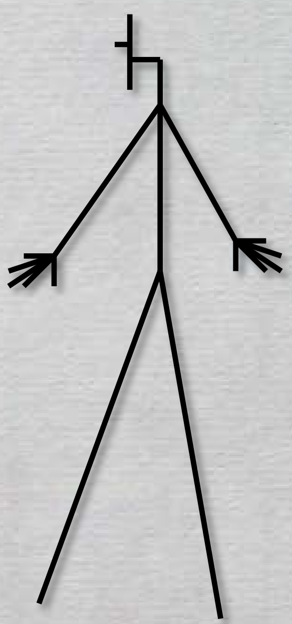

*CP: Parent and Children Crabs by Toshiyuki Meguro*



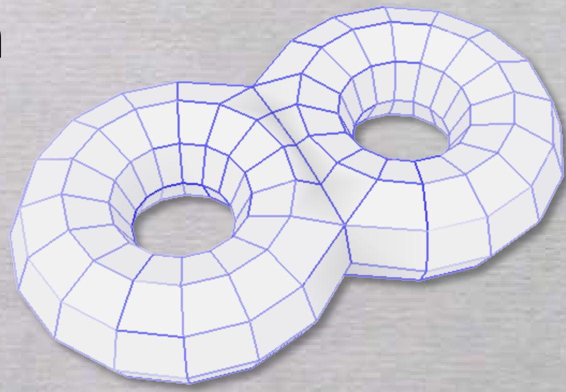
*Scutigera by Brian Chan*



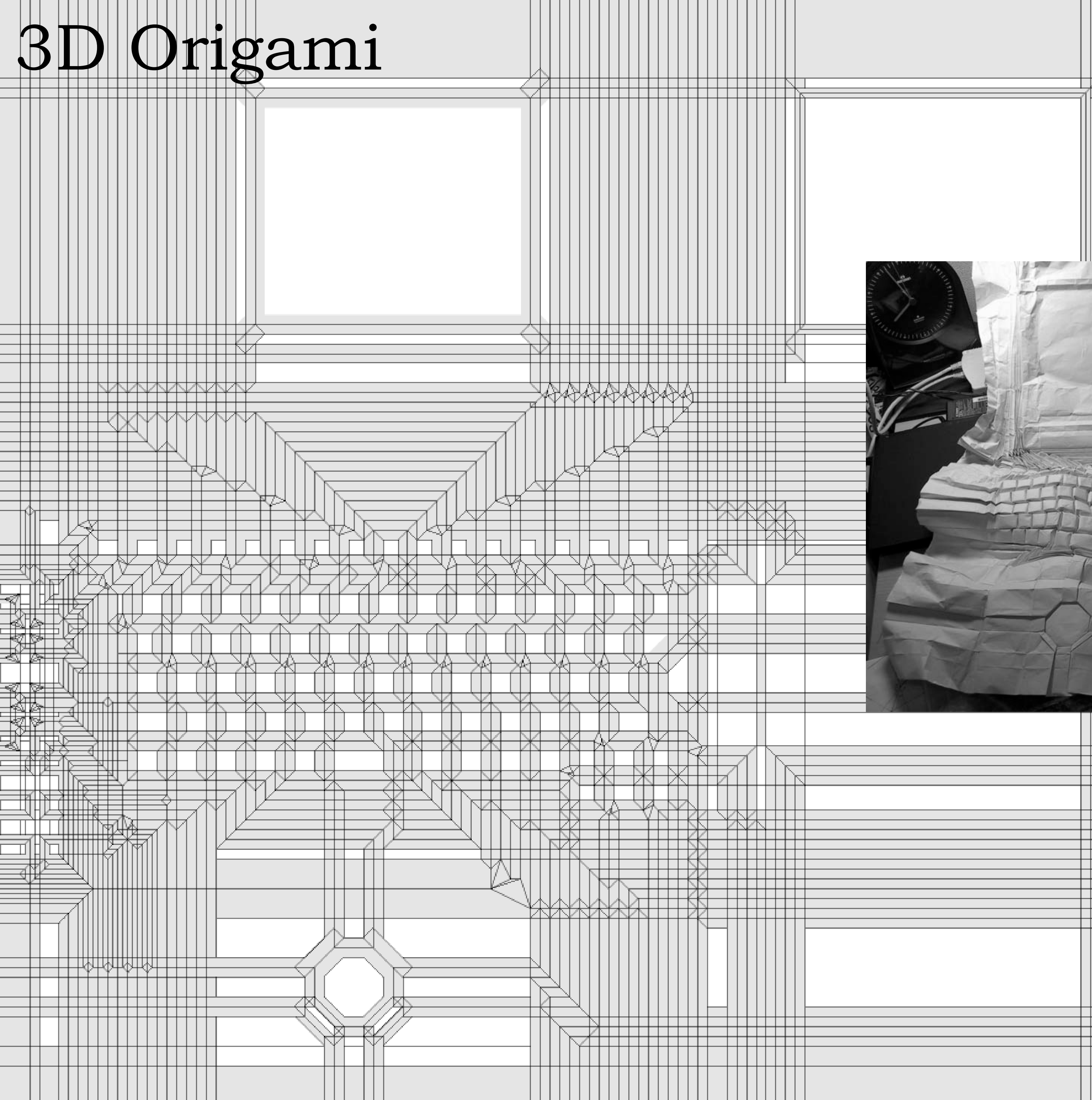
# 1D vs. 3D



- Circle River Method / Tree Method
  - Works fine for tree-like objects
  - Does not fit to 3D objects
- Origamizer / Freeform Origami
  - 3D Polyhedron, surface approximation
  - What You See Is What You Fold



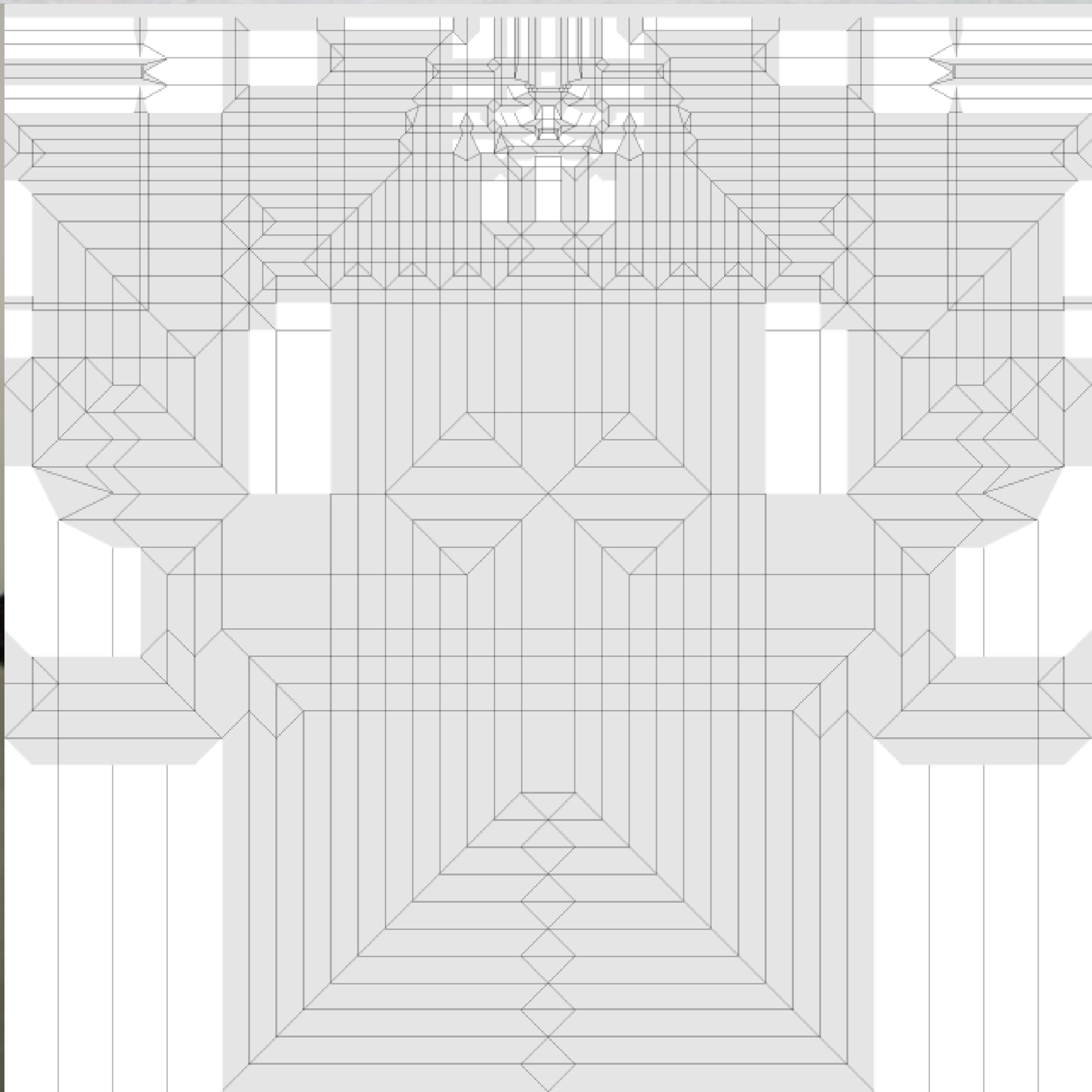
# 3D Origami



Laptop PC 2003
by Tomohiro Tachi
not completed

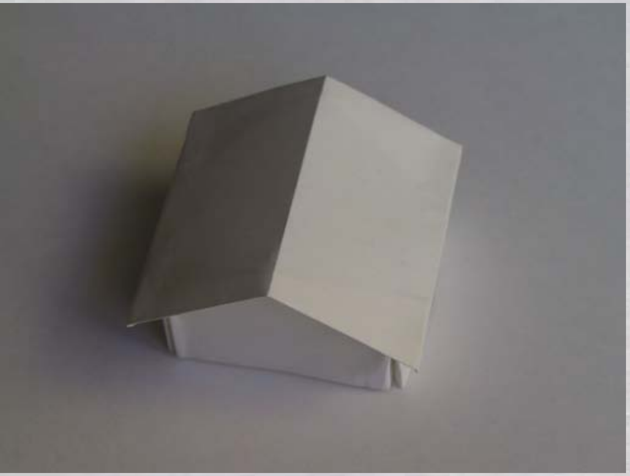
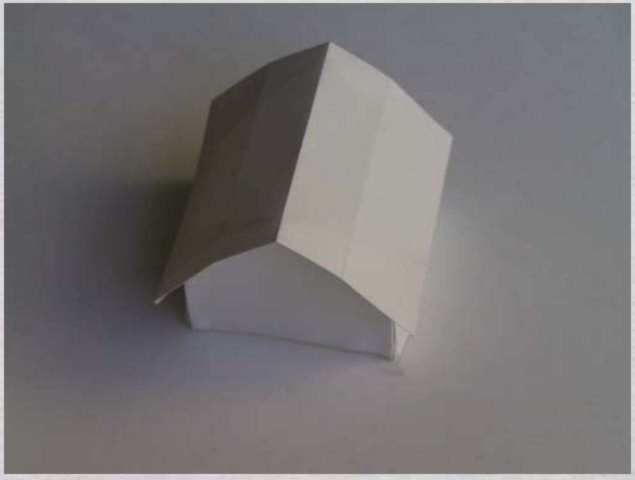
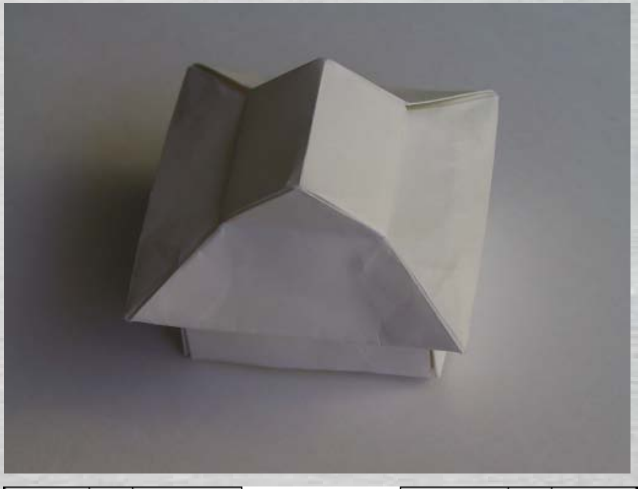
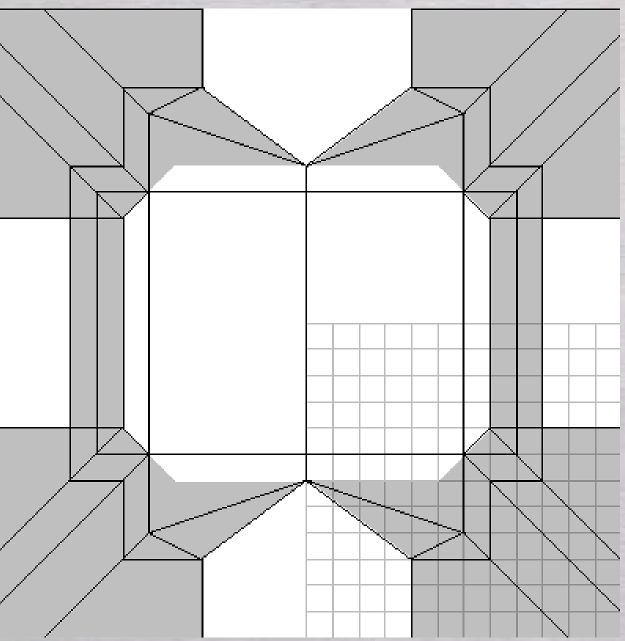
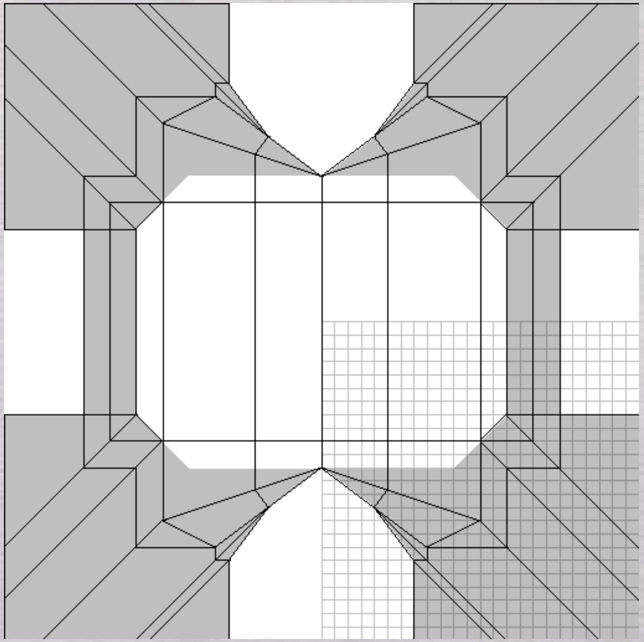
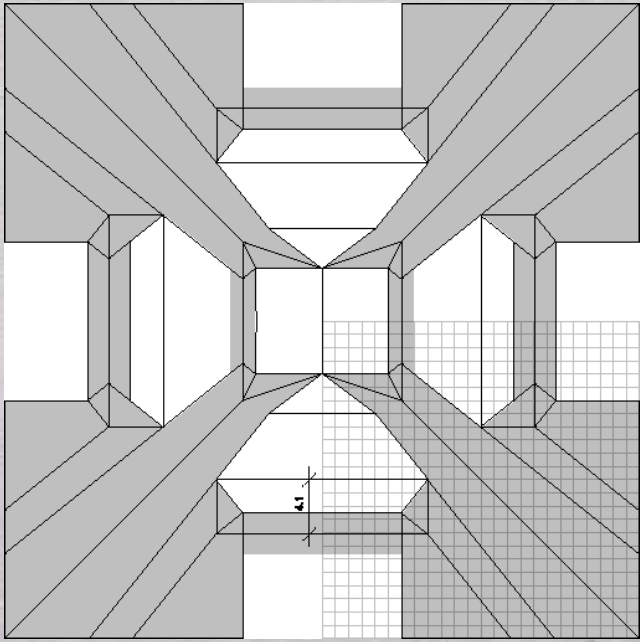
# 3D Origami

Human 2004



# 3D Origami

Roofs 2003

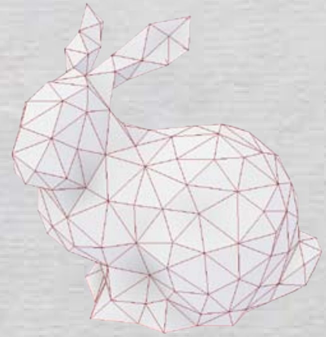







Everything seems to be possible!

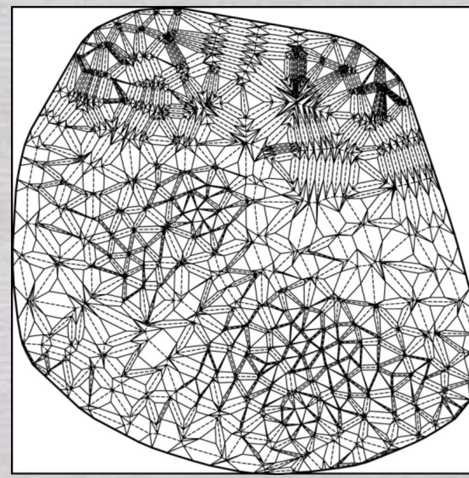
# Problem: realize arbitrary polyhedral surface with a developable surface

- Geometric Constraints
  - Developable Surf
  - Piecewise Linear
  - Forget about Continuous Folding Motion

- Potential Application
  - Fabrication by folding and bending



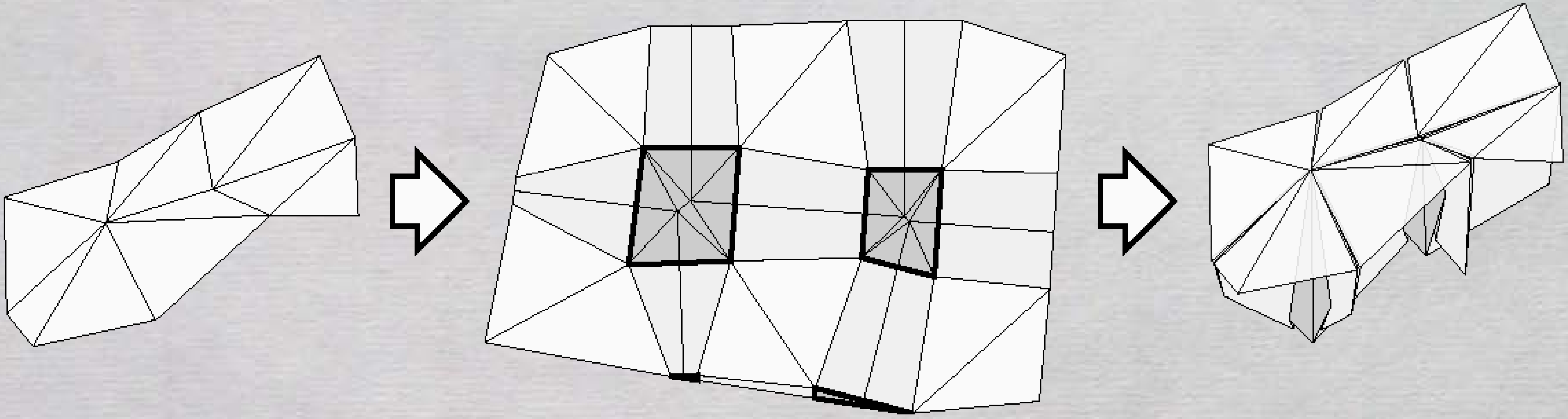
Input: Arbitrary Polyhedron



Output: Crease Pattern

Folded Polyhedron

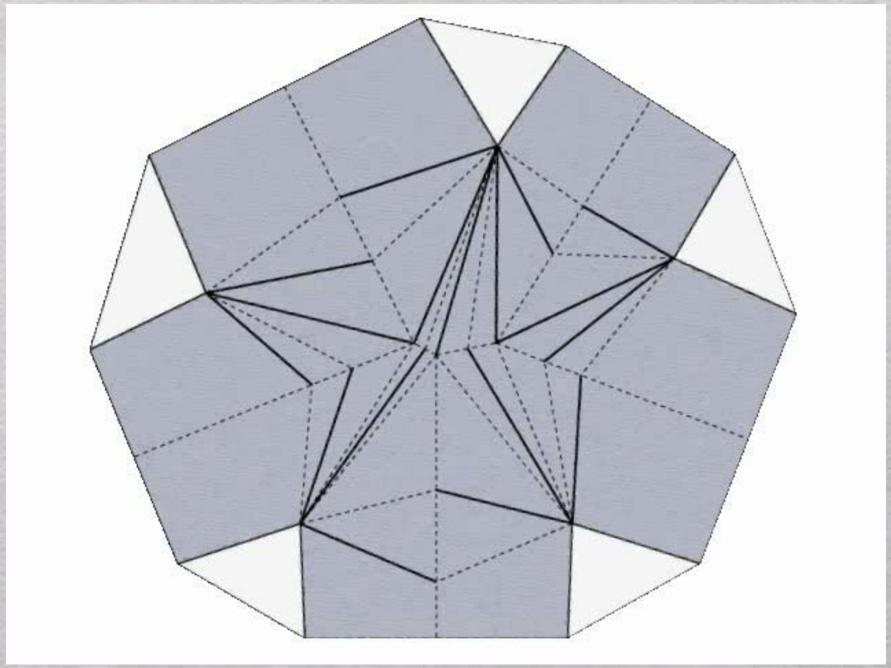
# Approach: Make "Tuck"



- Tuck develops into
  - a plane
- Tuck folds into
  - a flat state hidden behind polyhedral surface

→Important Advantage:

We can make Negative Curvature Vertex



# Basic Idea

Origamize Problem

↓

Lay-outing Surface Polygons Properly

↓

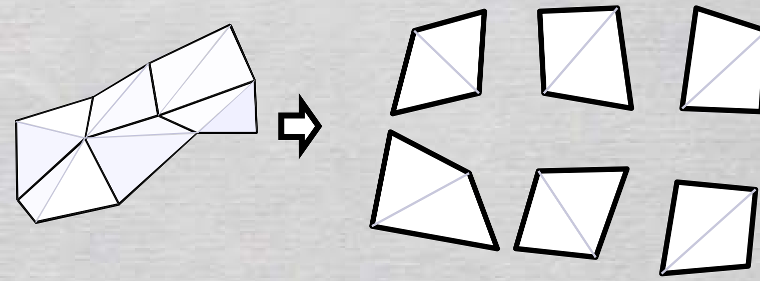
Tessellating Surface Polygons and "Tucking Molecules"

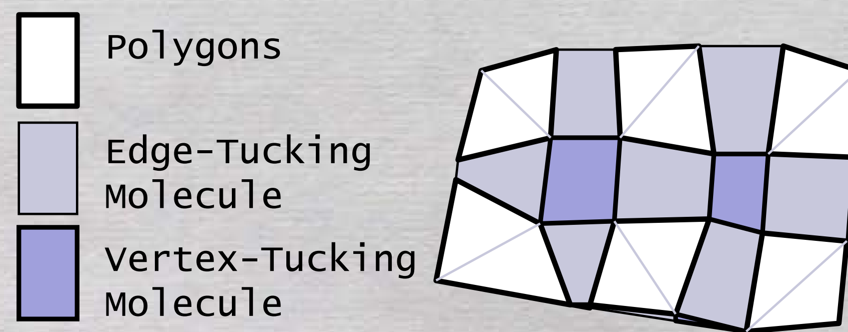
↓

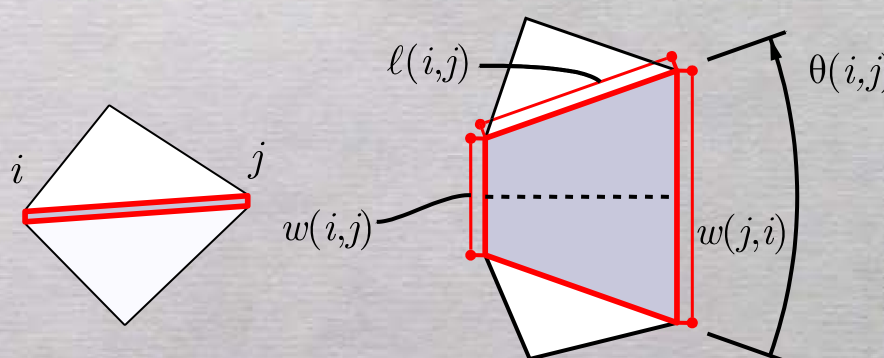
Parameter everything by Tucking Molecule:

- Angle $\theta(i, j)$
- Distance $w(i, j)$

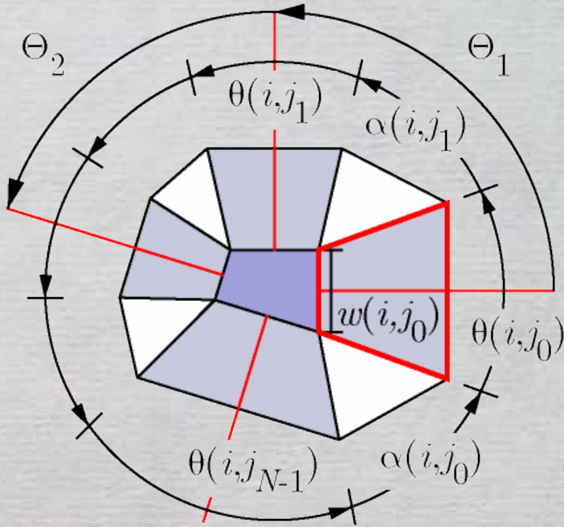
$$\theta(j, i) = -\theta(i, j)$$
$$w(j, i) = w(i, j) + 2\lambda(i, j)\sin(0.5\theta(i, j))$$





# Geometric Constraints (Equations)




$$\sum_{n=0}^{N-1}\theta(i,j_n)=2\pi-\sum_{n=0}^{N-1}\alpha(i,j_n) \qquad \cdots(1)$$

$$\sum_{n=0}^{N-1}w(i,j_n)\begin{bmatrix}\cos\left(\sum_{m=1}^{n}\Theta_m\right)\\ \sin\left(\sum_{m=1}^{n}\Theta_m\right)\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix} \qquad \cdots(2)$$

where $\Theta_m=\frac{1}{2}\theta(i,j_{m-1})+\alpha(i,j_m)+\frac{1}{2}\theta(i,j_m)$

Two-Step Linear Mapping

1. Mapping based on (1) (linear)

$$\mathbf{C_w w}=\mathbf{b}$$

2. Mapping based on (2) (linear)

$$\mathbf{w}=\mathbf{C}_w^{+}\mathbf{b}+\left(\mathbf{I}_{N_{edge}}-\mathbf{C}_w^{+}\mathbf{C}_w\right)\mathbf{w}_0$$

where $\mathbf{C}_w^{+}$ is the generalized inverse of $\mathbf{C}_w$

If the matrix is full-rank, $\mathbf{C}_w^{+}=\mathbf{C}_w^{\mathrm{T}}\left(\mathbf{C}_w\mathbf{C}_w^{\mathrm{T}}\right)^{-1}$

gives ($N_{edge}$-2$N_{vert}$)dimensional solution space

(within the space, we solve the inequalities)

# Geometric Constraints (Inequalities)

- 2D Cond.
  - Convex Paper $\theta(i,o) \geq \pi$
    $w(i,o) \geq 0$
  - Non-intersection $-\pi < \theta(i,j) < \pi$
    $\min(w(i,j), w(j,i)) \geq 0$
    $0 \leq \Theta_m < \pi$
  - Crease pattern non-intersection

$$\phi(i,j) \leq \arctan \frac{2\ell(i,j)\cos\frac{1}{2}\theta(i,j)}{w(i,j)+w(j,i)} + 0.5\pi$$

- 3D Cond.

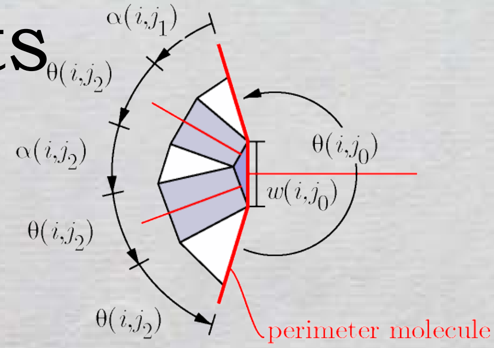
  for tuck proxy angle $\tau'(i,j)$ and depth $d'(i)$

  - Tuck angle condition

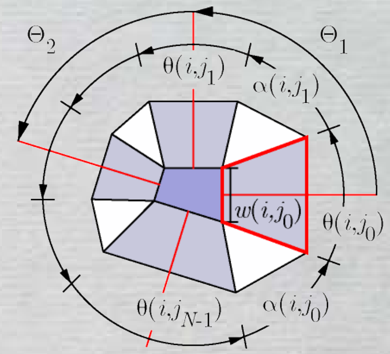
$$\phi(i,j) - \frac{1}{2}\theta(i,j) \leq \pi - \tau'(i,j)$$

  - Tuck depth condition

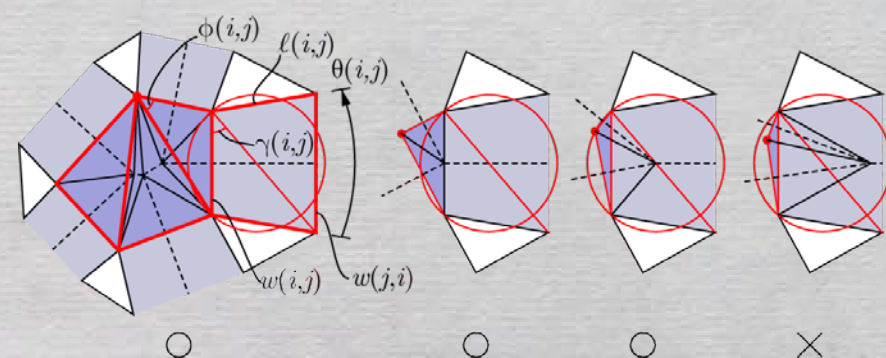
$$w(i,j) \leq 2\sin\left(\tau'(i,j) - \frac{1}{2}\alpha(i,j)\right)d'(i)$$



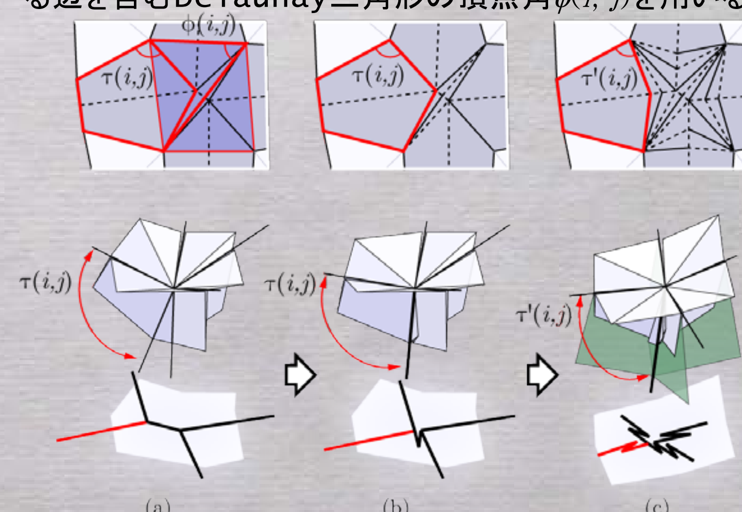

Convexity of paper



Non intersection (convexity of molecule)

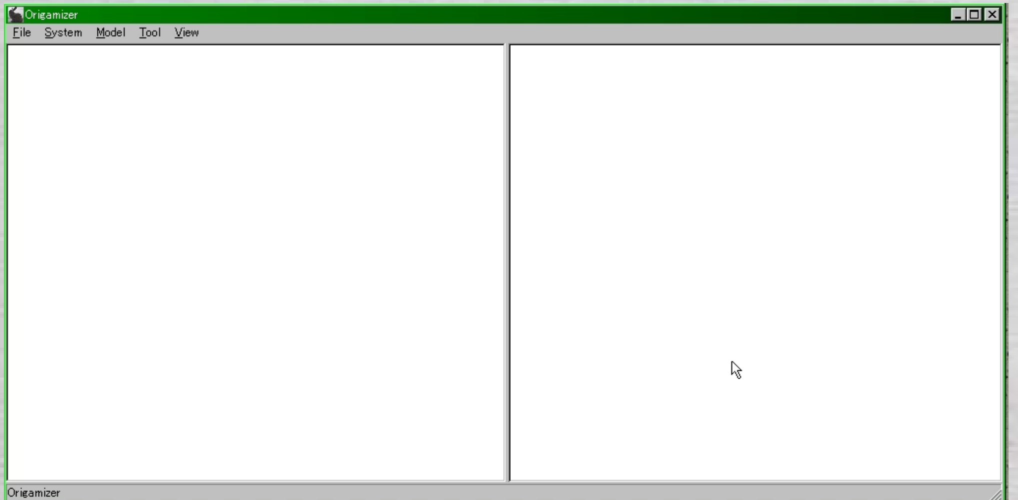


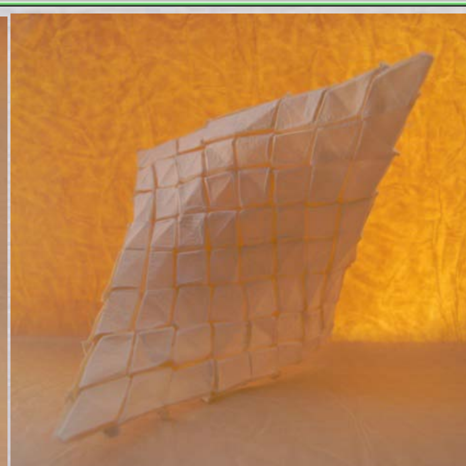

展開図の妥当条件： 頂点襞分子$i$と稜線襞分子$ij$が共有する辺を含むDelaunay三角形の頂点角$\phi(i, j)$を用いる。



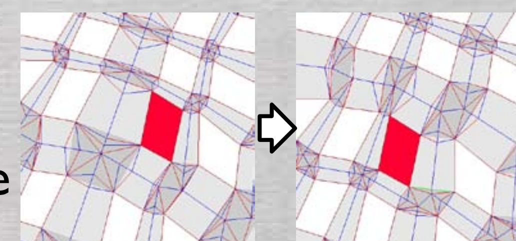
# Design System: Origamizer

3D CP

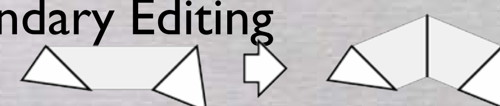


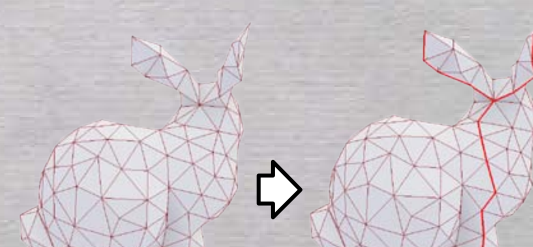



- Auto Generation of Crease Pattern
- Interactive Editing (Search within the solution space)
  - Dragging Developed Facets
  - Edge

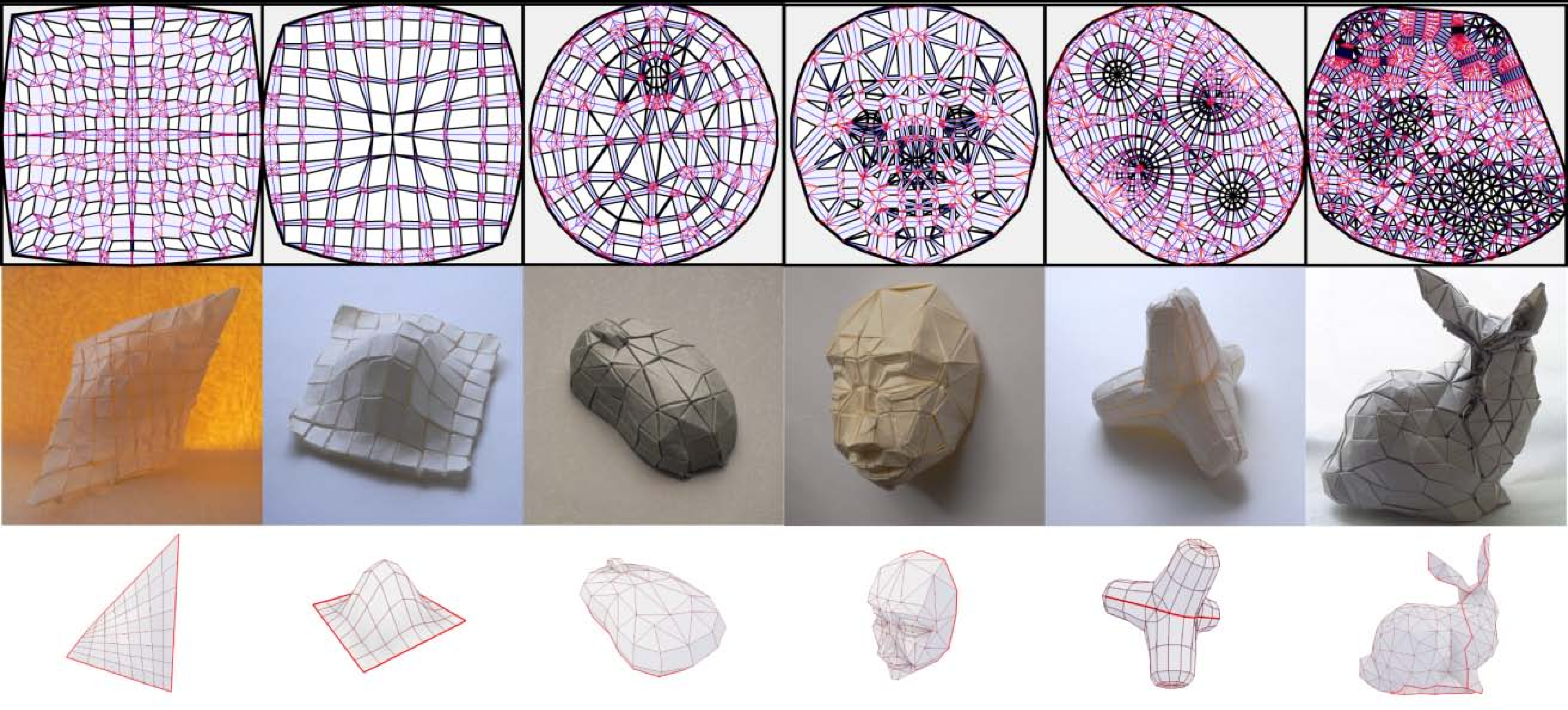


  - Boundary Editing



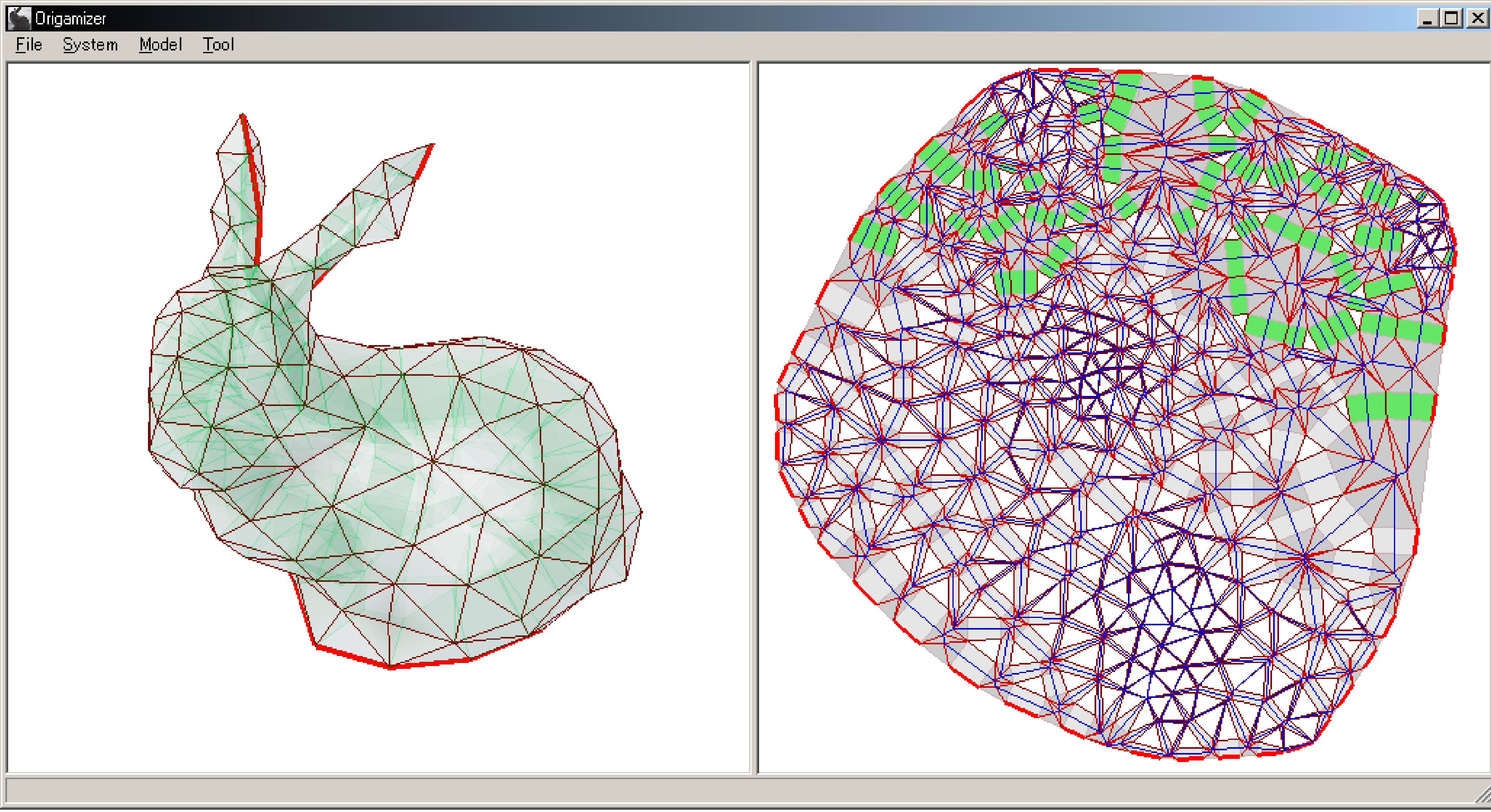


Developed in the project
“3D Origami Design Tool”
of IPA ESPer Project

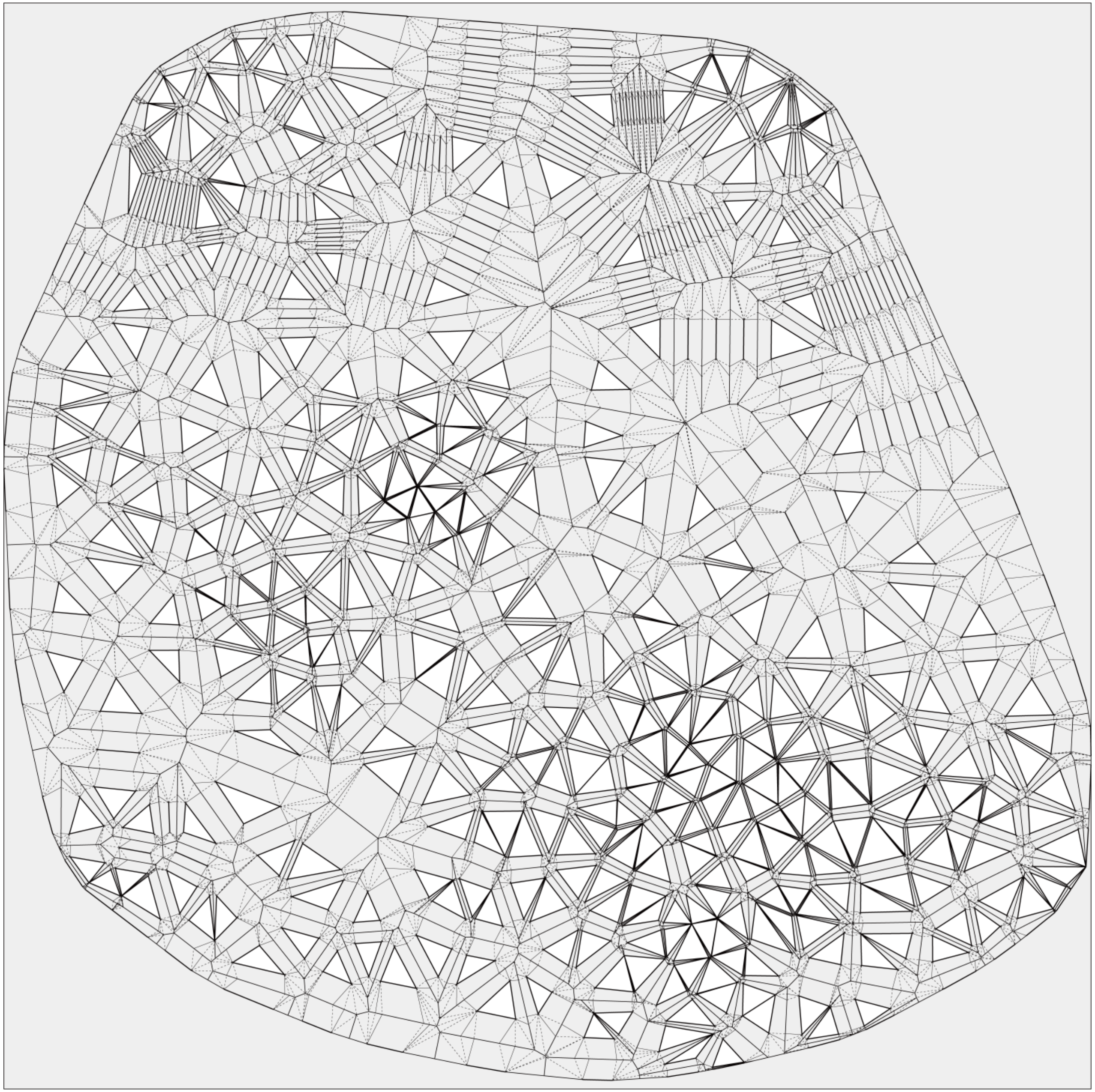


(a)Hyperbolic Paraboloid (b)Gaussian (c)Mouse (d)Mask (e)Tetrapod (f)Stanford Bunny

# How to Fold Origami Bunny




0. Get a crease pattern using Origamizer

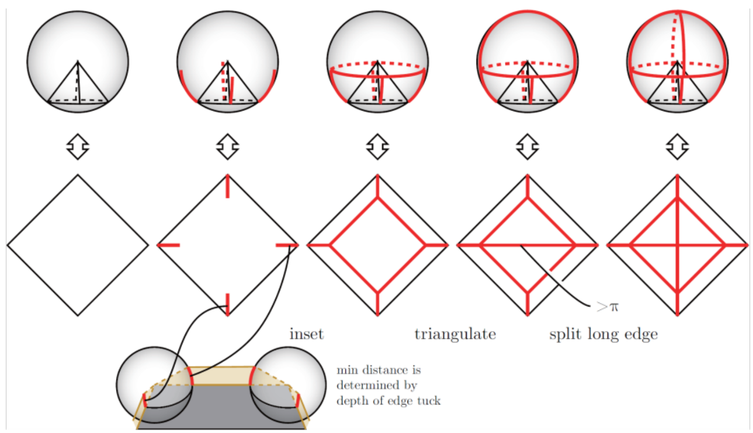


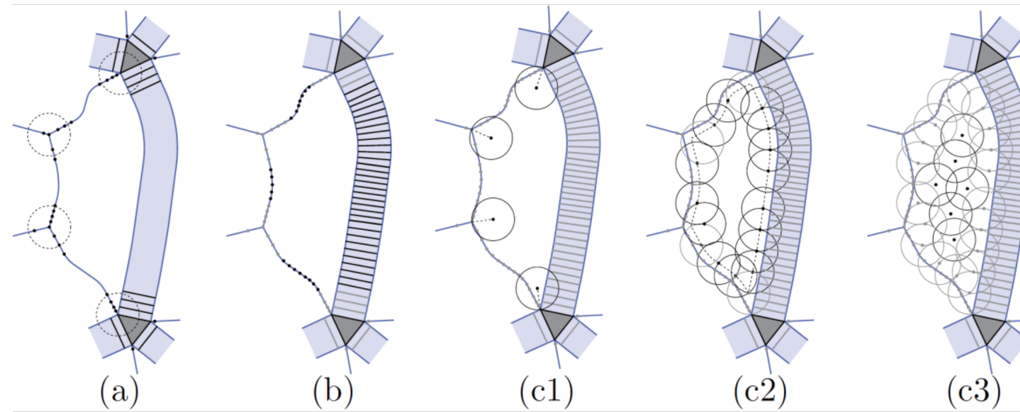
1. Fold Along the Crease Pattern

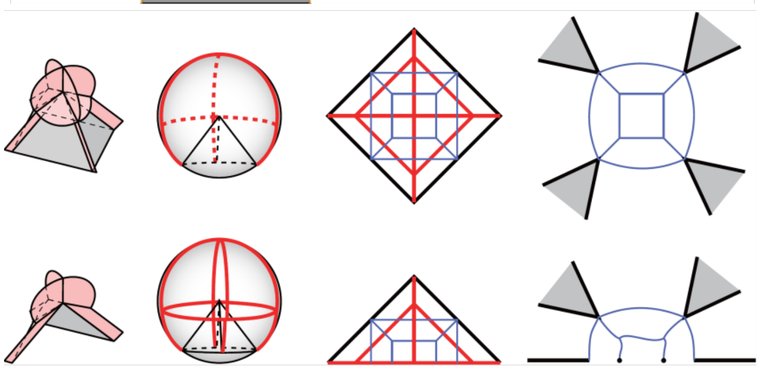
2. Done!

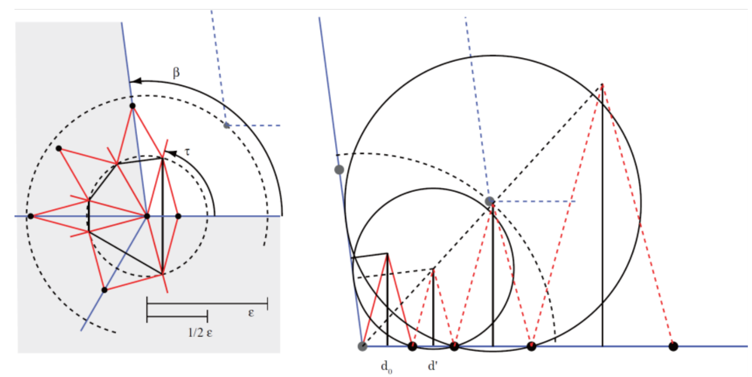
# Proof?

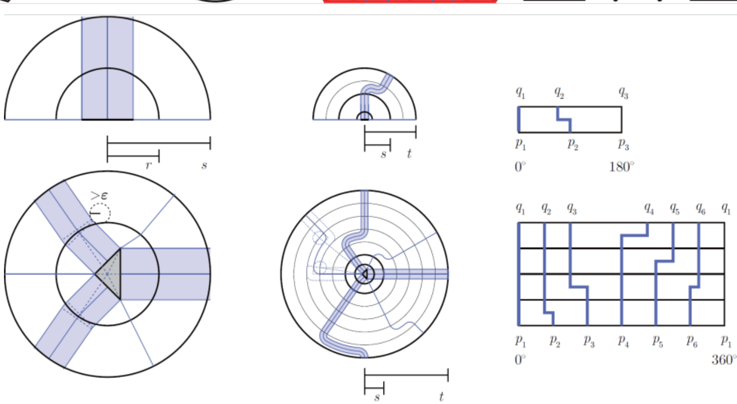
# Ongoing joint work with Erik Demaine

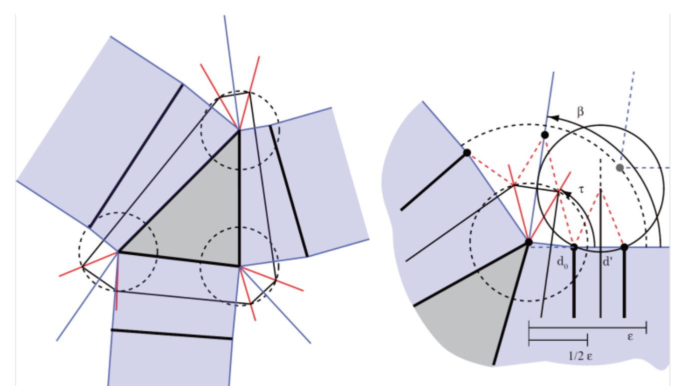




# 2

# Freeform Origami

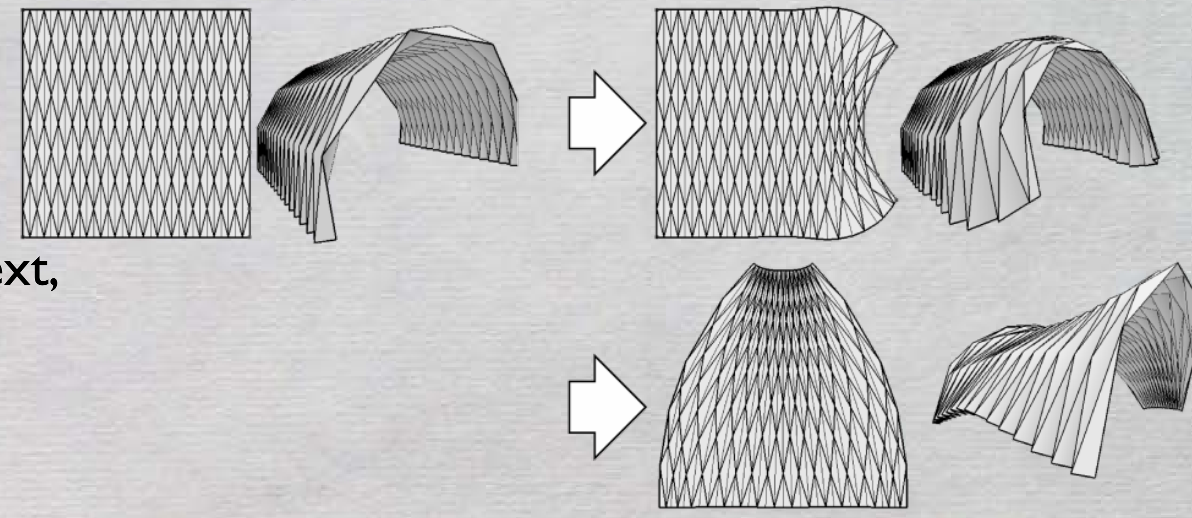
Related Papers:

• Tomohiro Tachi, "Freeform Variations of Origami", in Proceedings of The 14th International Conference on Geometry and Graphics (ICGG 2010), Kyoto, Japan, pp. 273-274, August 5-9, 2010.
(to appear in Journal for Geometry and Graphics Vol. 14, No. 2)

• Tomohiro Tachi: "Smooth Origami Animation by Crease Line Adjustment ," ACM SIGGRAPH 2006 Posters, 2006.

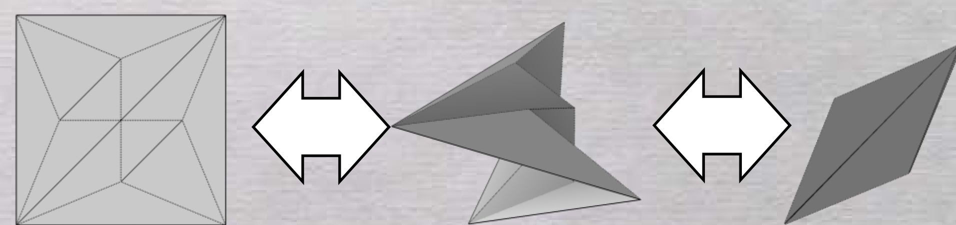
# Objective of the Study

1. freeform
   - Controlled 3D form
   - Fit function, design context, preference, ...

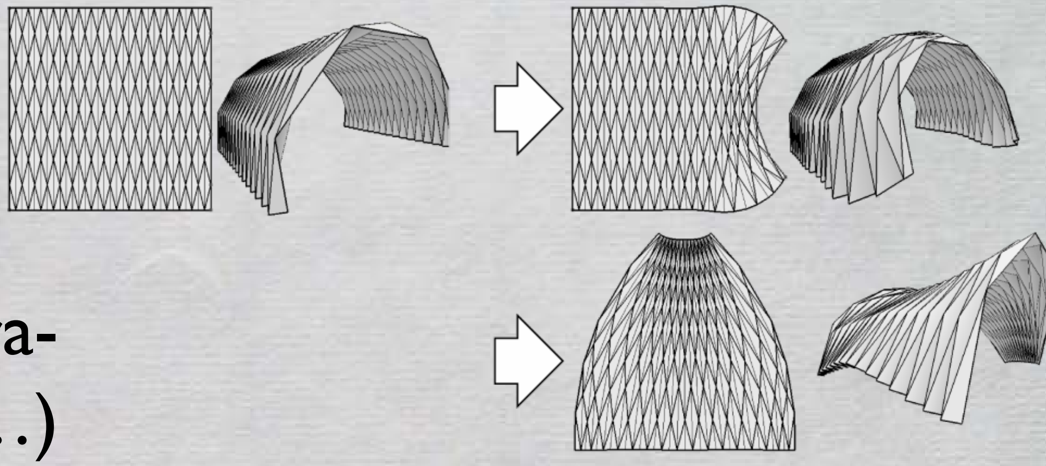


2. origami
   utilize the properties
   - Developability
     → Manufacturing from a sheet material
     based on Folding, Bending
   - Flat-foldability
     → Folding into a compact configuration
     or Deployment from 2D to 3D
   - Rigid-foldability
     → Transformable
     Structure
   - Elastic Properties

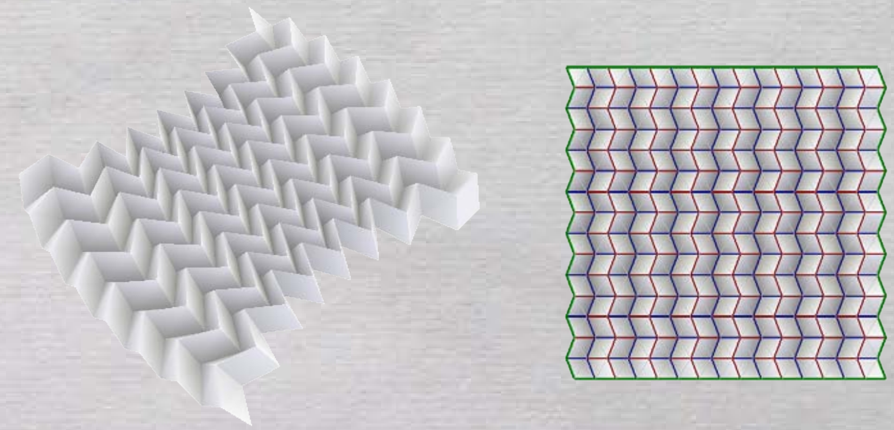
   ...

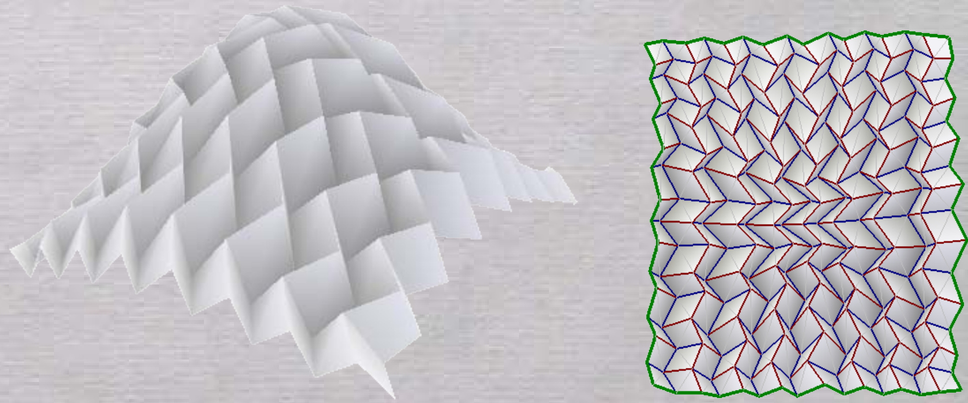


# Proposing Approach



- Initial State: existing origami models (e.g. Miura-ori, Ron Resch Pattern, …) + Perturbation consistent with the origami conditions.
- Straightforward user interface.

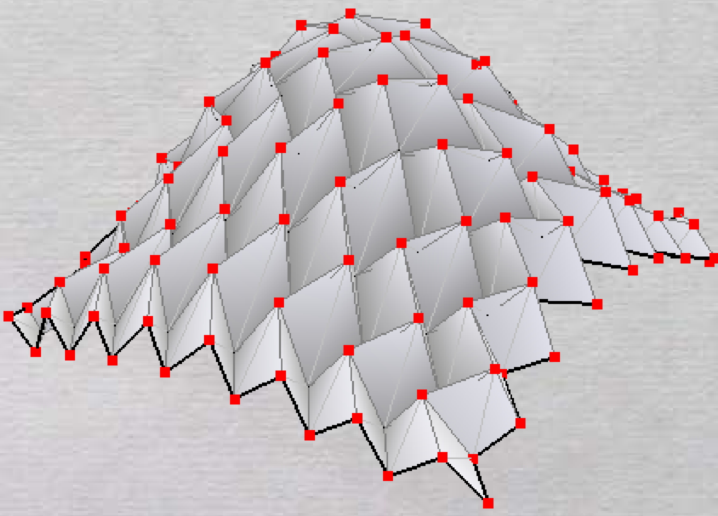




# Model

- Triangular Mesh (triangulate quads)
- Vertex coordinates represent the configuration
  - $3N_v$ variables, where $N_v$ is the # of vertices

$$\mathbf{X} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_{N_v} \\ y_{Nv} \\ z_{Nv} \end{bmatrix}$$



- The configuration is constrained by developability, flat-foldability, …

# Developability

Engineering Interpretation

→ Manufacturing from a sheet material
based on Folding, Bending

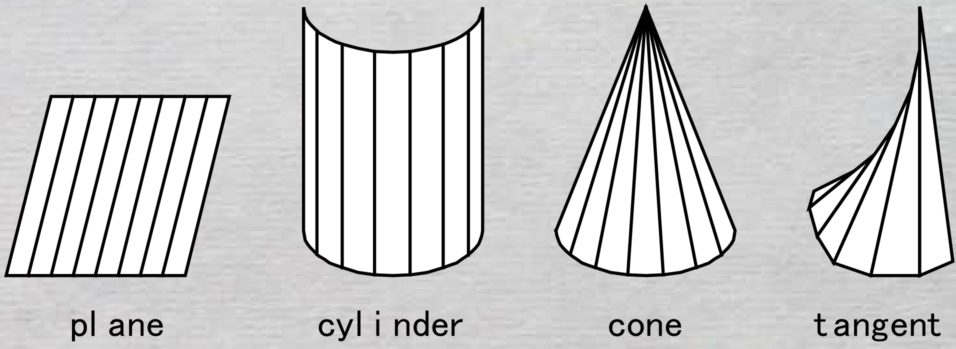
- Global condition
  - There exists an isometric map to a plane.

⇔(if topological disk)

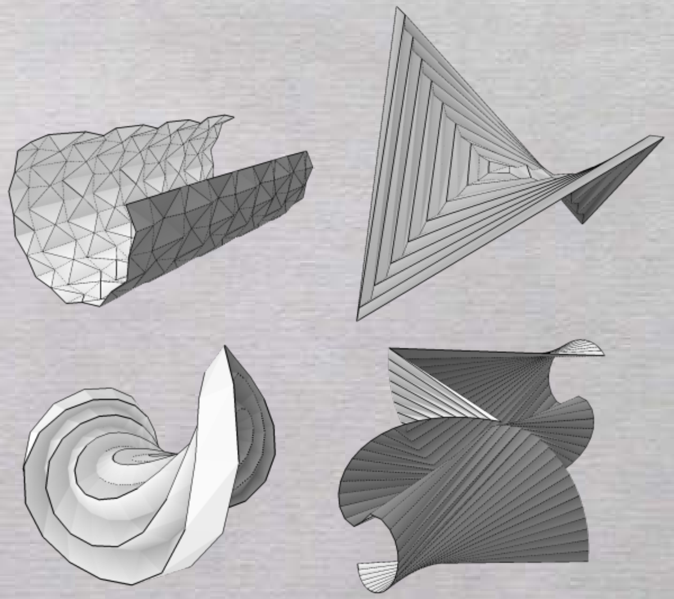
- Local condition
  - Every point satisfies
    - Gauss curvature = 0

# Developable Surface

- Smooth Developable Surface
  - $G^2$ surface (curvature continuous)
    - "Developable Surface" (in a narrow sense)
    - Plane, Cylinder, Cone, Tangent surface
  - $G^1$ Surface (smooth, tangent continuous)
    - "Uncreased flat surface"
    - piecewise Plane, Cylinder, Cone, Tangent surface

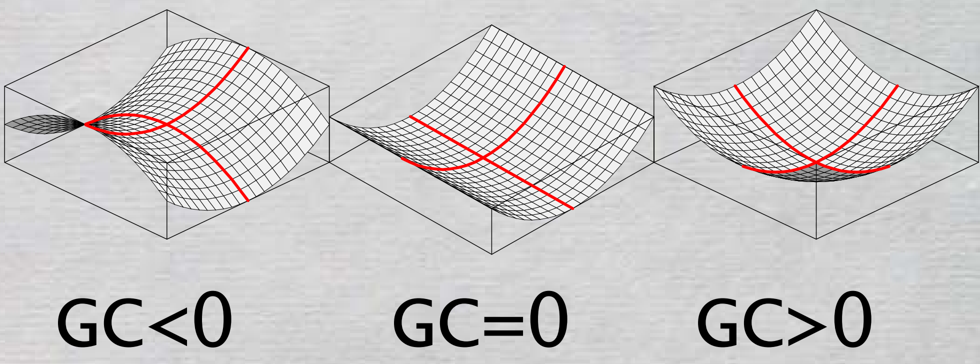



- Origami
  - $G^0$ Surface
  - piecewise $G^1$ Developable $G^0$ Surface

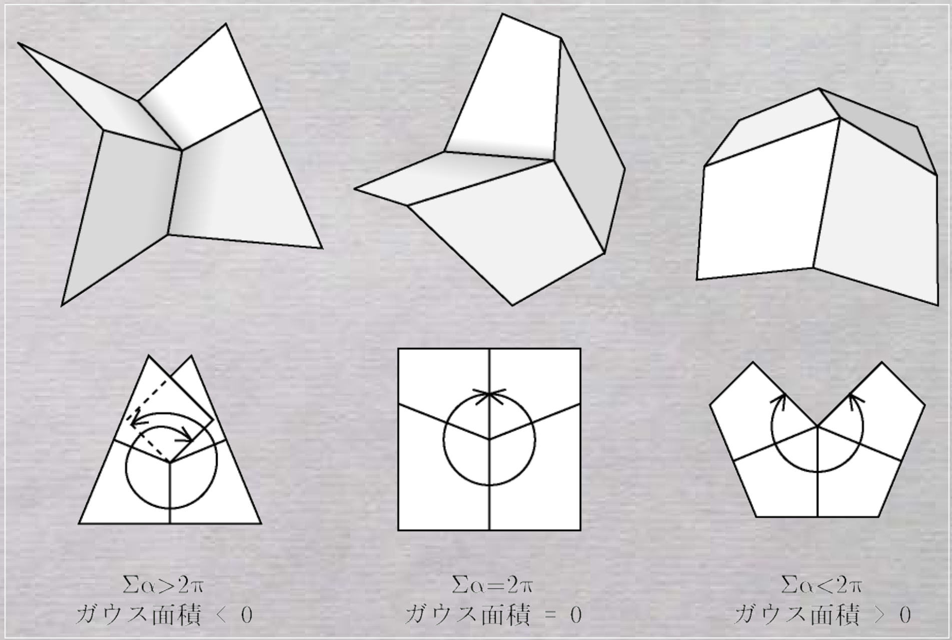


# Developability condition to be used

- Constraints
  - For every interior vertex v ($k_v$-degree), gauss area equals 0.




$$\mathbf{G}_v = 2\pi - \sum_{i=0}^{kv} \theta_i = 0$$



# Flat-foldability

Engineering Interpretation

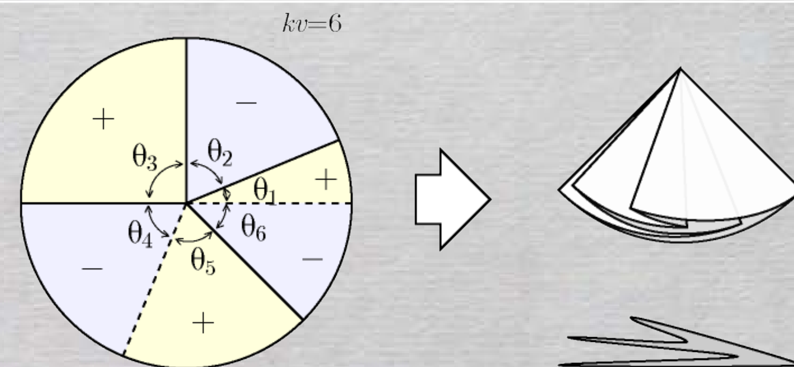
→ Folding into a compact configuration
or Deployment from 2D to 3D

- Isometry condition
  - isometric mapping with mirror reflection
- Layering condition
  - valid overlapping ordering
    - globally : NP Complete [Bern and Hayes 1996]

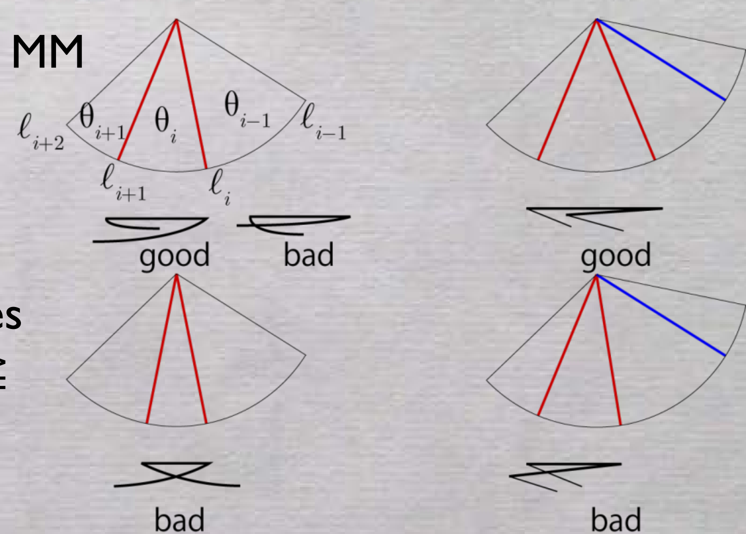
# Flat-foldability condition to be used

- Isometry
  - ⇔ Alternating sum of angles is 0 [Kawasaki 1989]

$$\mathbf{F}_v = \sum_{i=0}^{kv} \mathrm{sgn}(i)\theta_i = 0$$




- Layering
  - ⇒ [kawasaki 1989]
    - If $\theta_i$ is between foldlines assigned with MM or VV, $\theta_i \geqq \min(\theta_{i-1}, \theta_{i+1})$
  - + empirical condition [tachi 2007]
    - If $\theta_i$ and $\theta_{i+1}$ are composed byfoldlines assigned with MMV or VVM then, $\theta_i \geqq \theta_{i+1}$



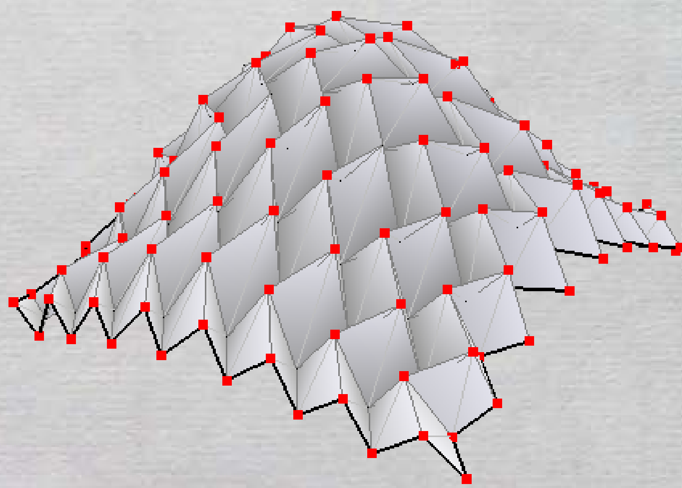
# Other Conditions

- Conditions for fold angles
  - Fold angles $\rho$
  - V fold: $0<\rho<\pi$
  - M fold: $-\pi<\rho<0$
  - crease: $-\alpha\pi<\rho<\alpha\pi$ ($\alpha=0$: planar polygon)
- Optional Conditions
  - Fixed Boundary
    - Folded from a specific shape of paper
  - Rigid bars
  - Pinning

# Settings

- Initial Figure:
  - Symmetric Pattern
- Freeform Deformation
  - Variables (3$N_v$)
    - Coordinates **X**
  - Constraints (2$N_{v_in}$+$N_c$)
    - Developability
    - Flat-foldability
    - Other Constraints

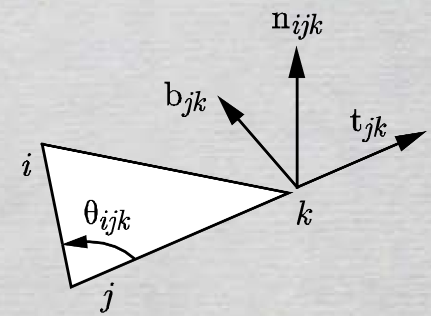
$$\mathbf{X} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_{N_v} \\ y_{Nv} \\ z_{Nv} \end{bmatrix}$$



Under-determined System

→Multi-dimensional Solution Space

# Solve Non-linear Equation




The infinitesimal motion satisfies:

$$\mathbf{C}\dot{\mathbf{X}} = \begin{bmatrix} \frac{\partial \mathbf{G}}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{F}}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{H}}{\partial \mathbf{X}} \end{bmatrix} \dot{\mathbf{X}} = \begin{bmatrix} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{G}}{\partial \boldsymbol{\rho}} \\ \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{F}}{\partial \boldsymbol{\rho}} \\ \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{H}}{\partial \boldsymbol{\rho}} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{X}} \\ \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{X}} \end{bmatrix} \dot{\mathbf{X}} = \mathbf{0}$$

$$\mathbf{G}_v = 2\pi - \sum_{i=0}^{kv} \theta_i = 0$$

$$\mathbf{F}_v = \sum_{i=0}^{kv} \operatorname{sgn}(i)\theta_i = 0$$

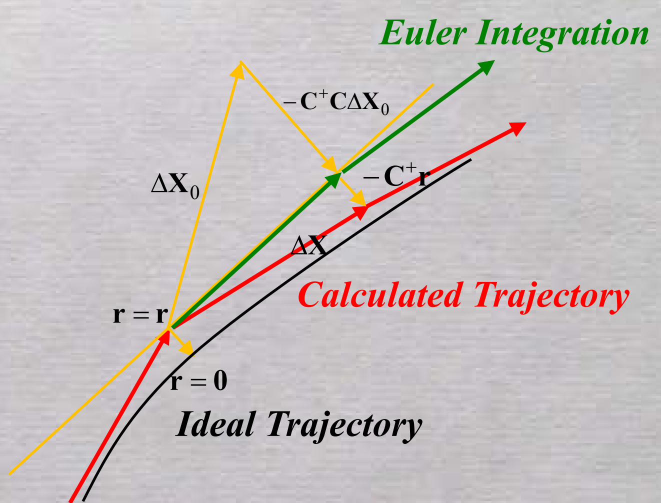
$$\frac{\partial \theta_{ijk}}{\partial \mathbf{x}_i} = -\frac{1}{\ell_{ij}} \mathbf{b}_{ij}^{\mathrm{T}}$$

$$\frac{\partial \theta_{ijk}}{\partial \mathbf{x}_j} = \frac{1}{\ell_{ij}} \mathbf{b}_{ij}^{\mathrm{T}} + \frac{1}{\ell_{jk}} \mathbf{b}_{jk}^{\mathrm{T}}$$

$$\frac{\partial \theta_{ijk}}{\partial \mathbf{x}_k} = -\frac{1}{\ell_{jk}} \mathbf{b}_{jk}^{\mathrm{T}}$$

For an arbitrarily given (through GUI)
Infinitesimal Deformation $\Delta X_0$

$$\Delta \mathbf{X} = -\mathbf{C}^{+}\mathbf{r} + \left(\mathbf{I}_{3N_v} - \mathbf{C}^{+}\mathbf{C}\right)\Delta \mathbf{X}_0$$



# Freeform Origami

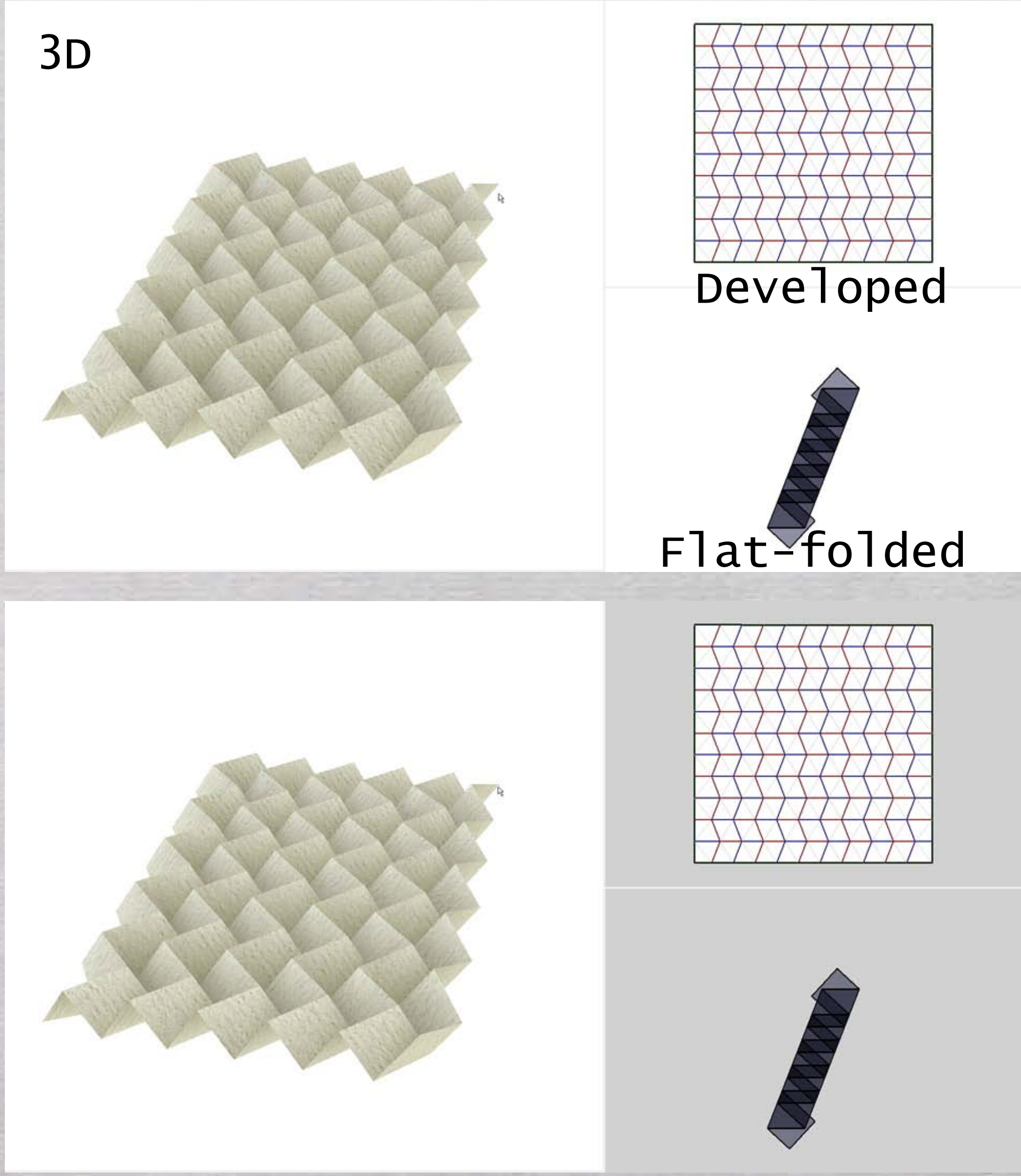
## Get A Valid Value

- Iterative method to calculate the conditions
- Form finding through User Interface

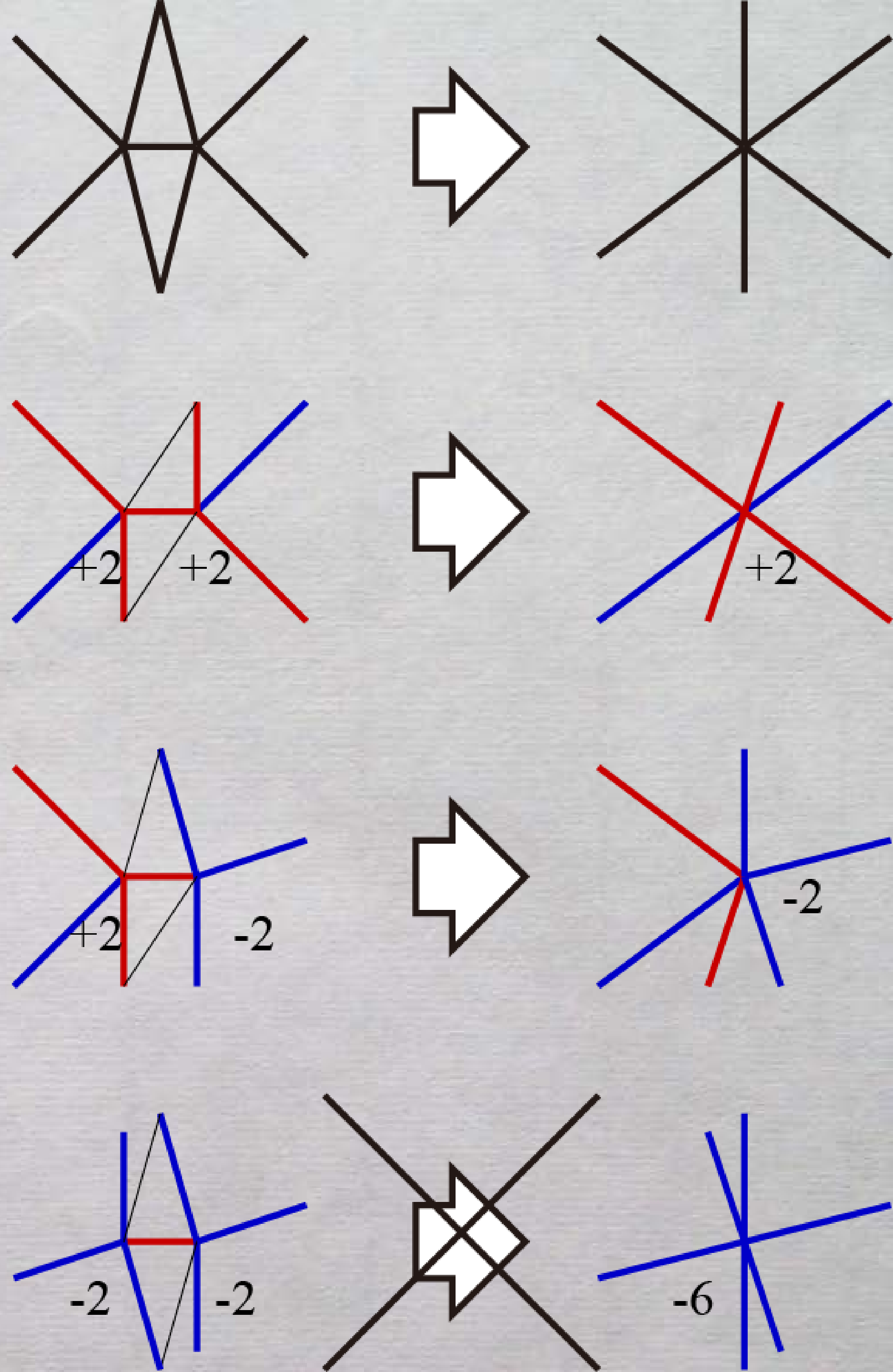
## Implementation

- Lang
  - C++, STL
- Library
  - BLAS (intel MKL)
- Interface
  - wxWidgets, OpenGL

## To be available on web



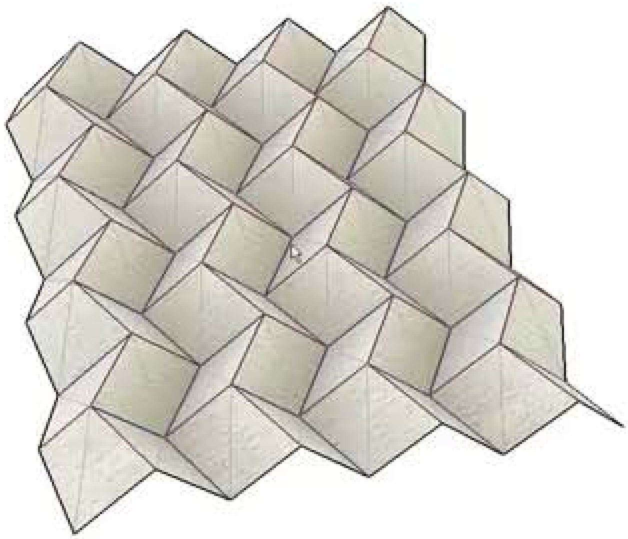
# Mesh Modification
# Edge Collapse

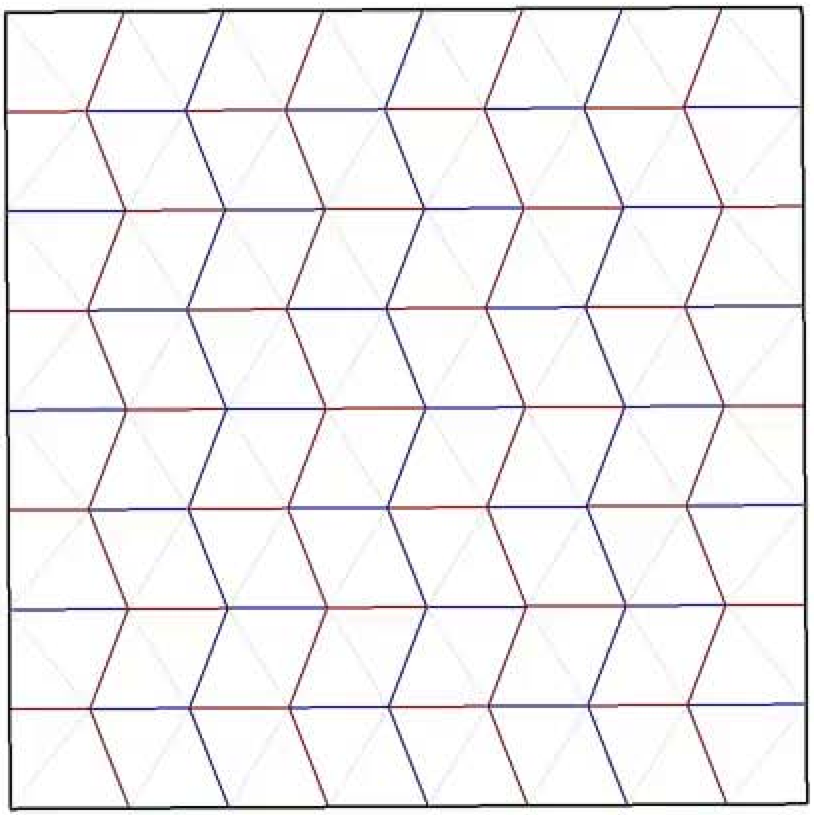



- Edge Collapse [Hoppe etal 1993]
- Maekawa's Theorem [1983] for flat foldable pattern

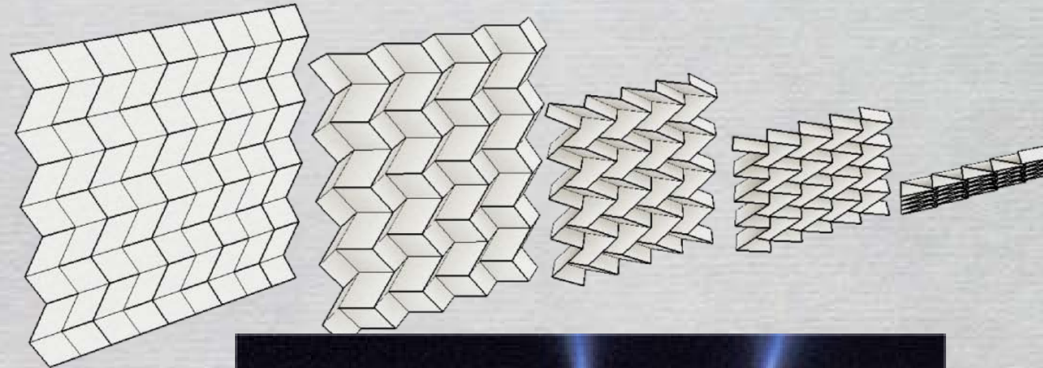
  M - V = ±2

# Mesh Modification





# Miura-Ori



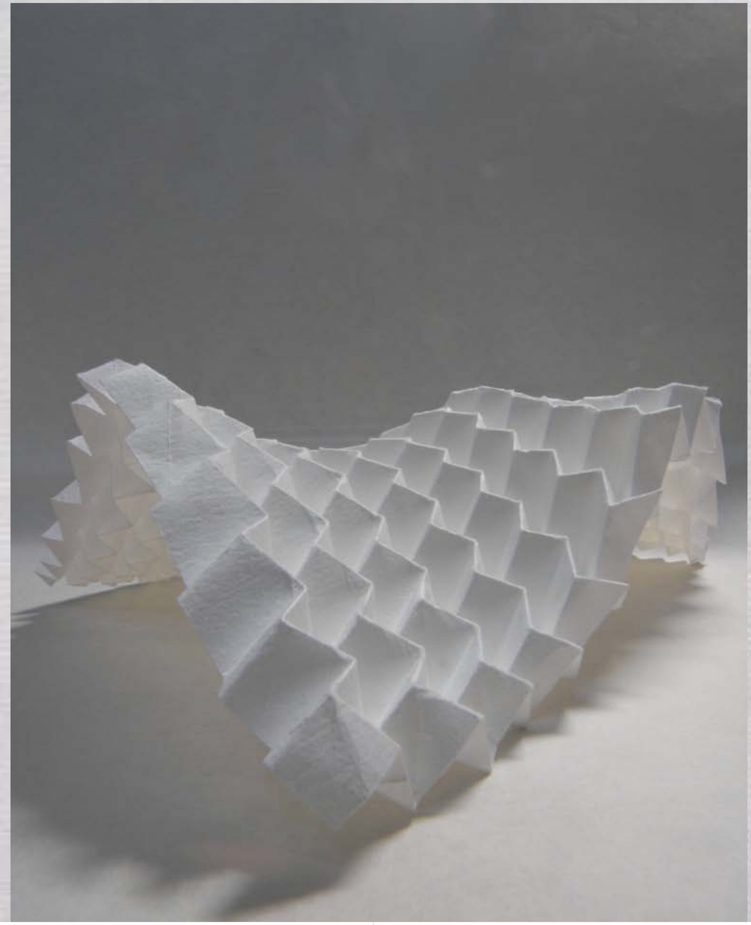
- Original
  - [Miura 1970]
- Application
  - bidirectionally expansible (one-DOF)
  - compact packaging
  - sandwich panel
- Conditions
  - Developable
  - Flat-foldable
  - op: (Planar quads)(→Rigid Foldable)

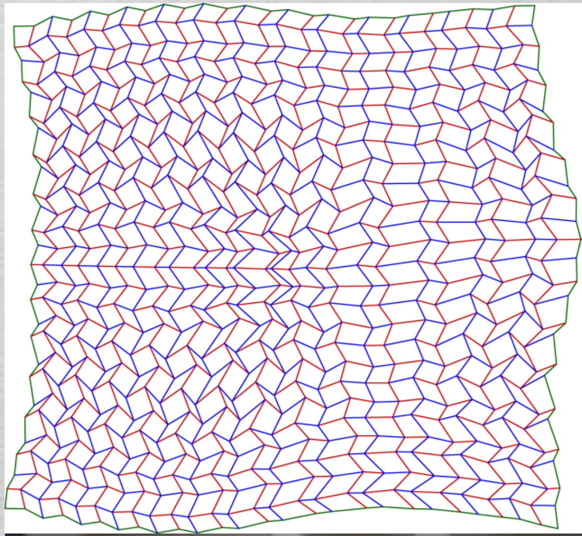
ISAS Space Flyer Unit

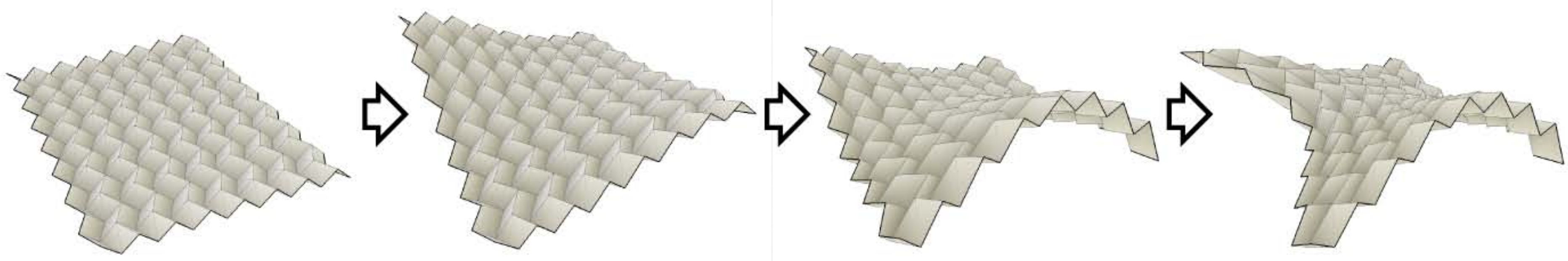
zeta core
[Koryo Miura 1972]

# Miura-ori Generalized

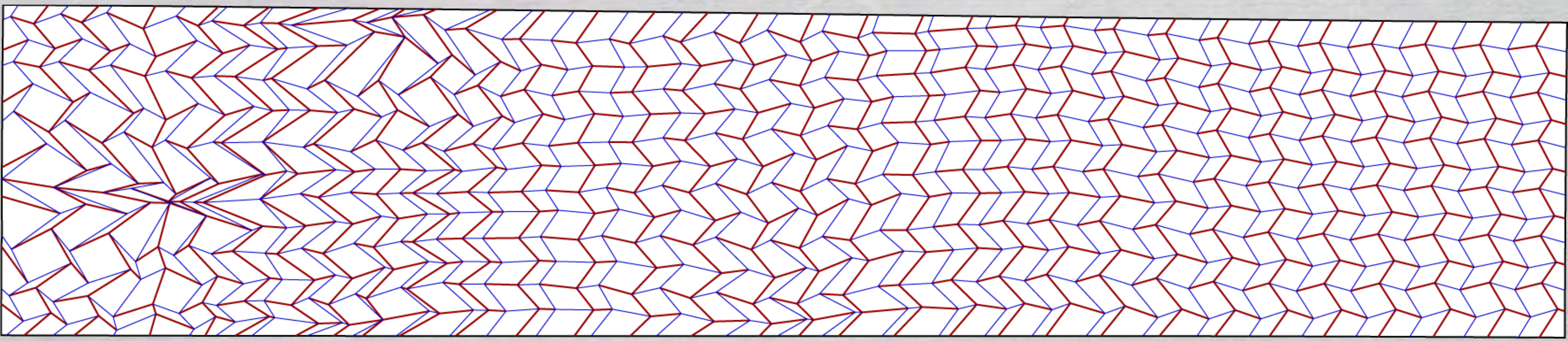
- Freeform Miura-ori

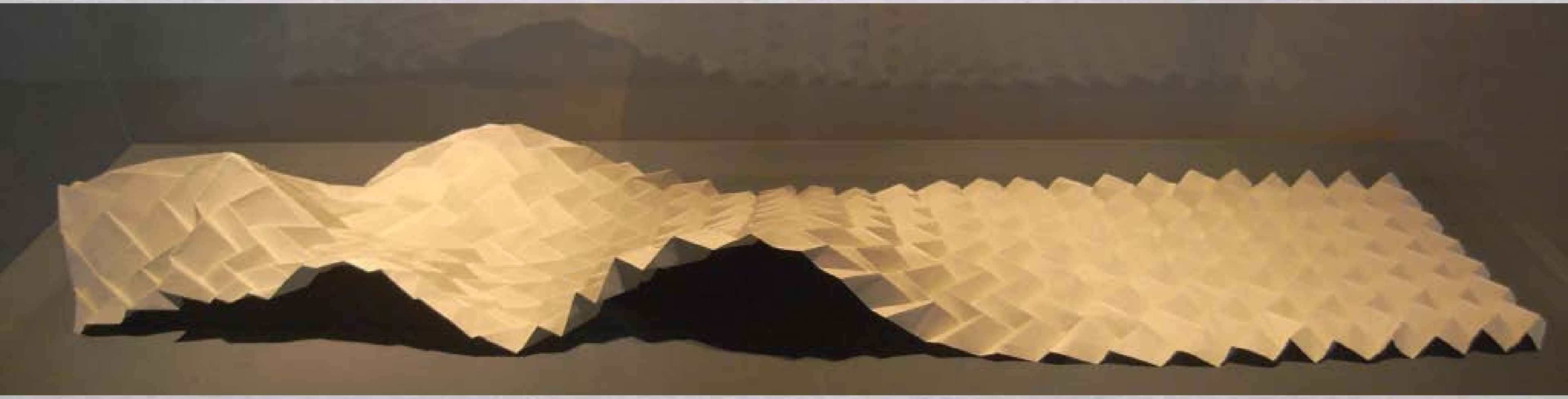






# Miura-ori Generalized

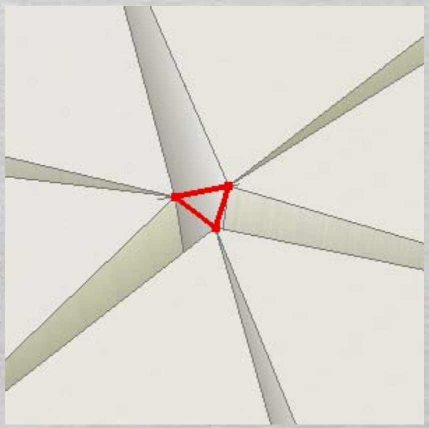




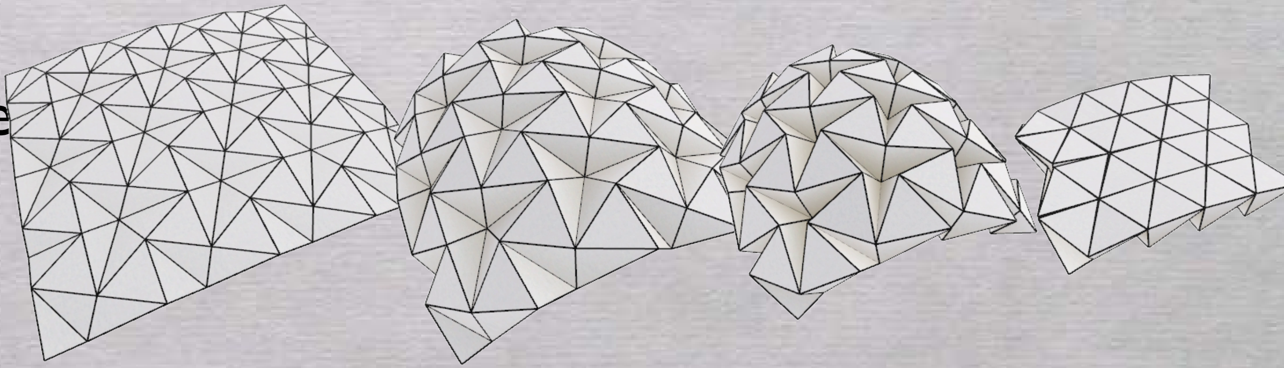
Origami Metamorphose Tomohiro Tachi 2010

# Ron Resch Pattern

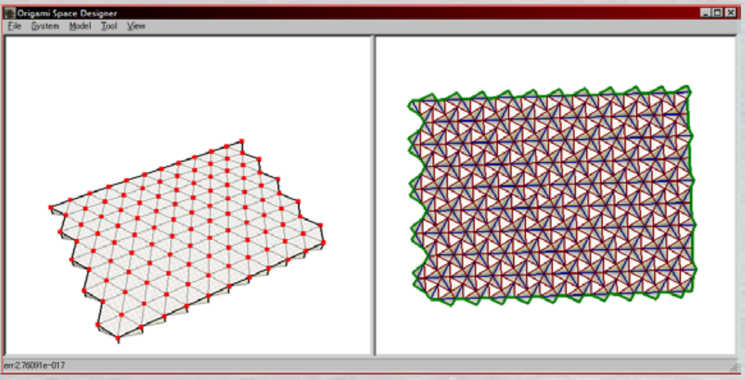
- Original
  - Resch [1970]
- Characteristics
  - Flexible (multiDOF)
  - Forms a smooth flat surface + scaffold

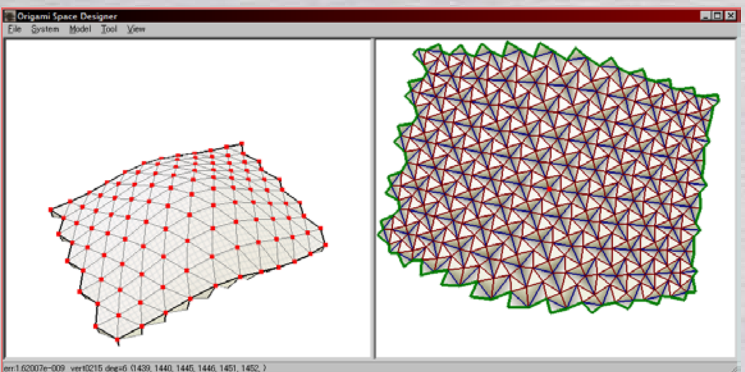


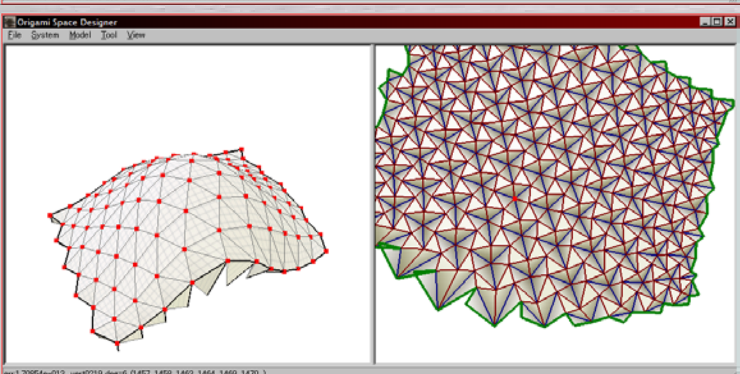
- Conditions
  - Developable
  - 3-vertex coincide

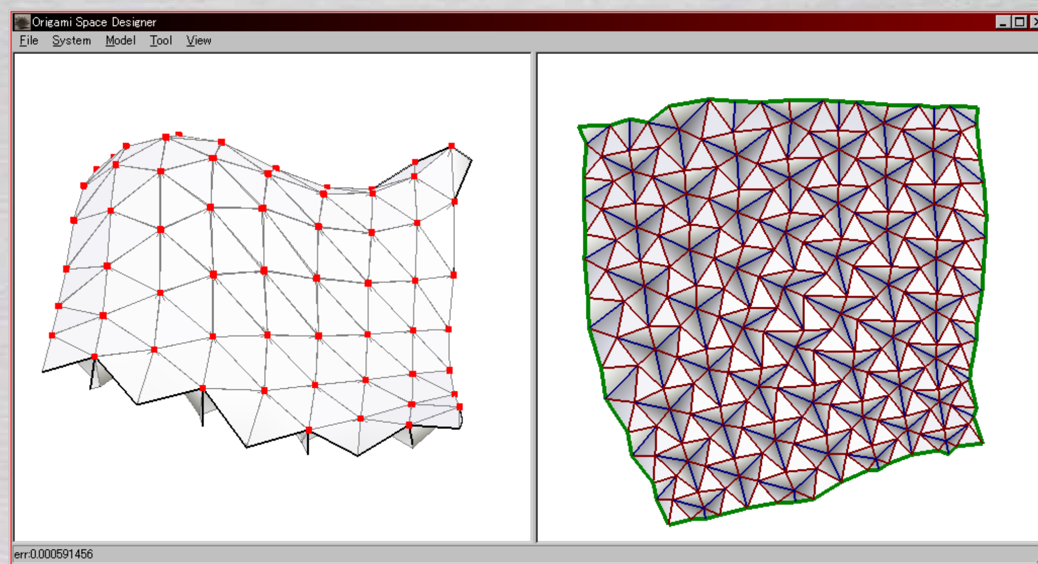


# Ron Resch Pattern Generalized

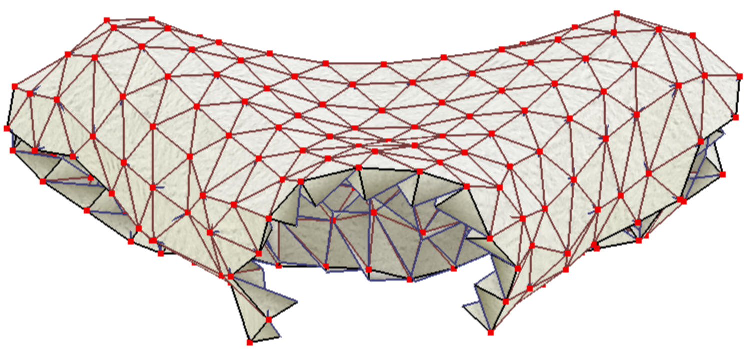


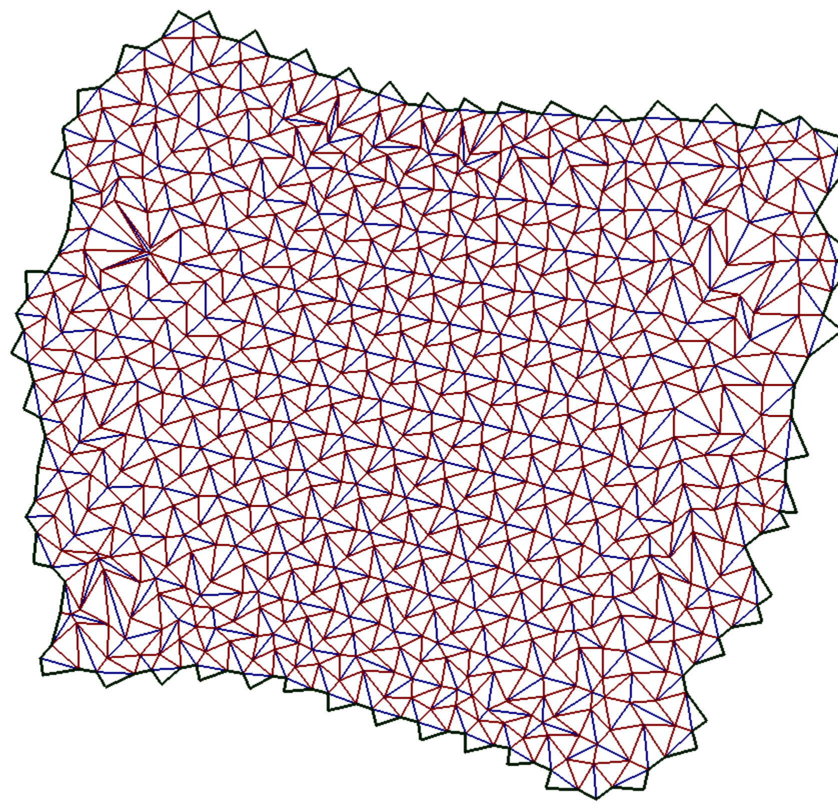



# Generalized Ron Resch Pattern

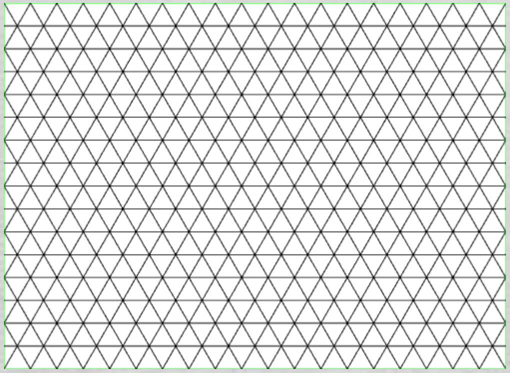
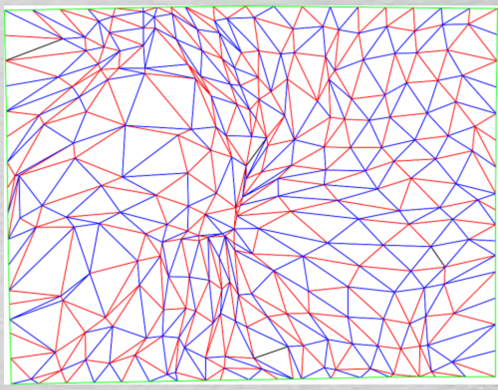
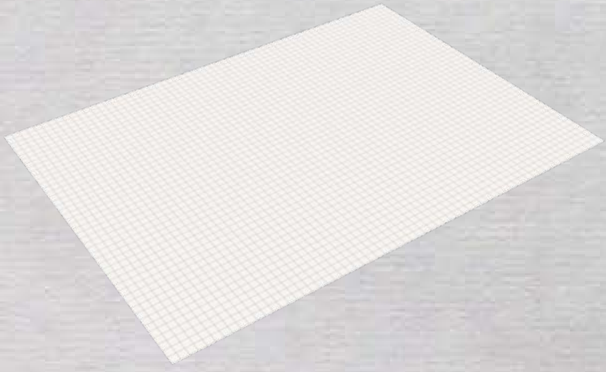


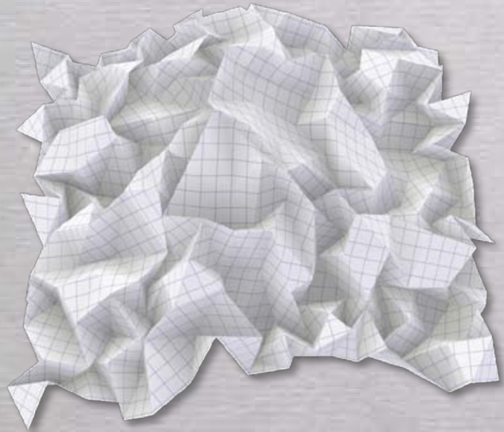


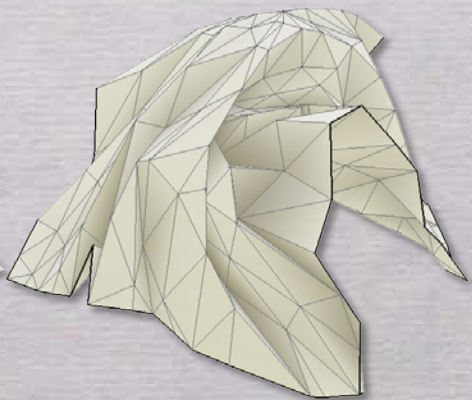
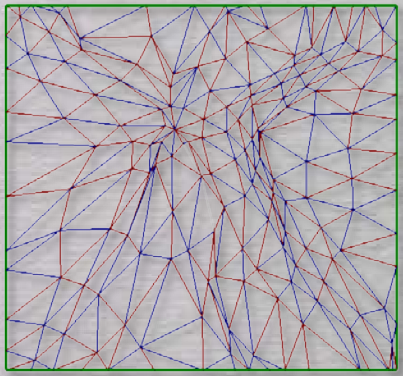
# Crumpled Paper

- Origami
  = crumpled paper
  = buckled sheet

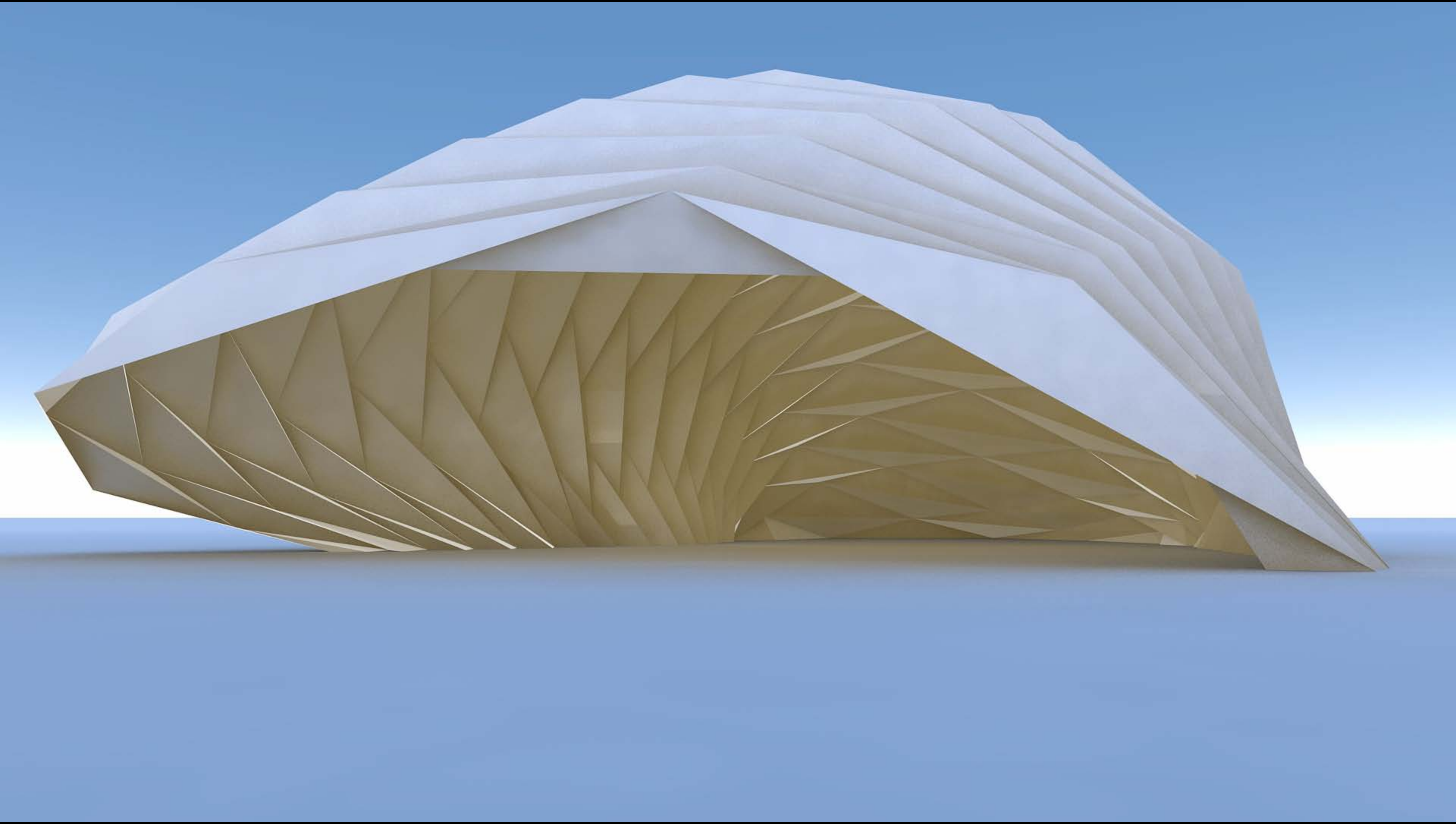
- Conditions
  - Developable
  - Fixed Perimeter



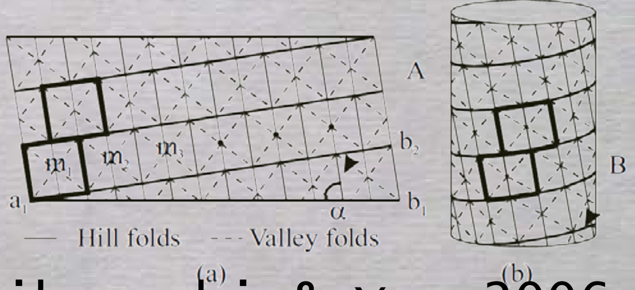
# crumpled paper example



# Waterbomb Pattern

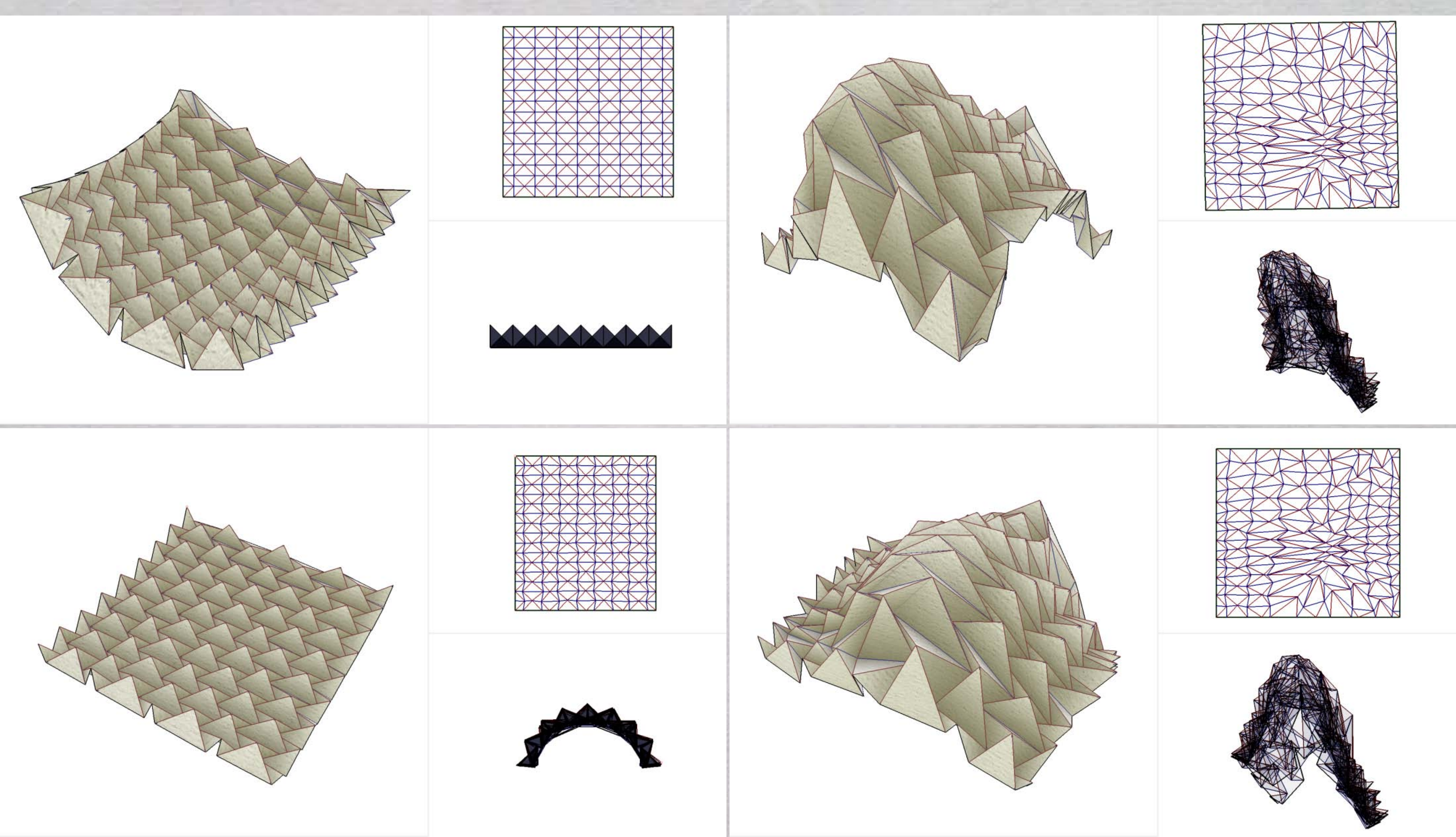
- "Namako" (by Shuzo Fujimoto)
- Characteristics
  - Flat-foldable
  - Flexible(multi DOF)
  - Complicated motion
- Application
  - packaging
  - textured material
  - cloth folding...

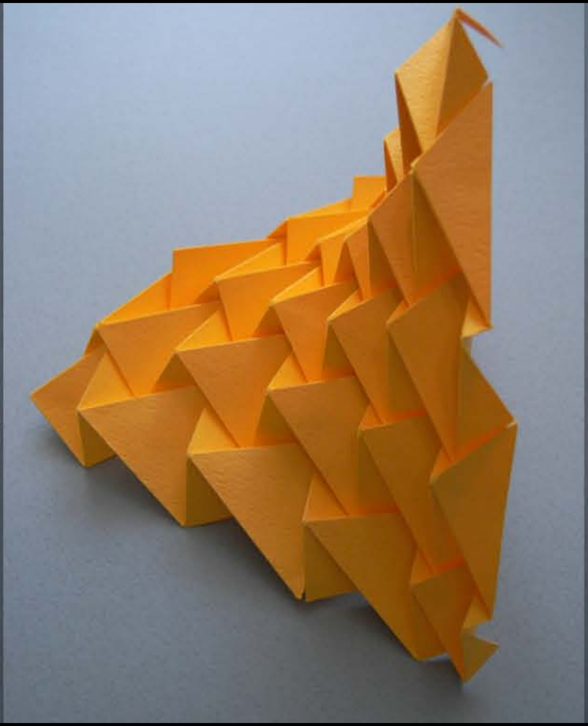
S. Mabona "Fugu"

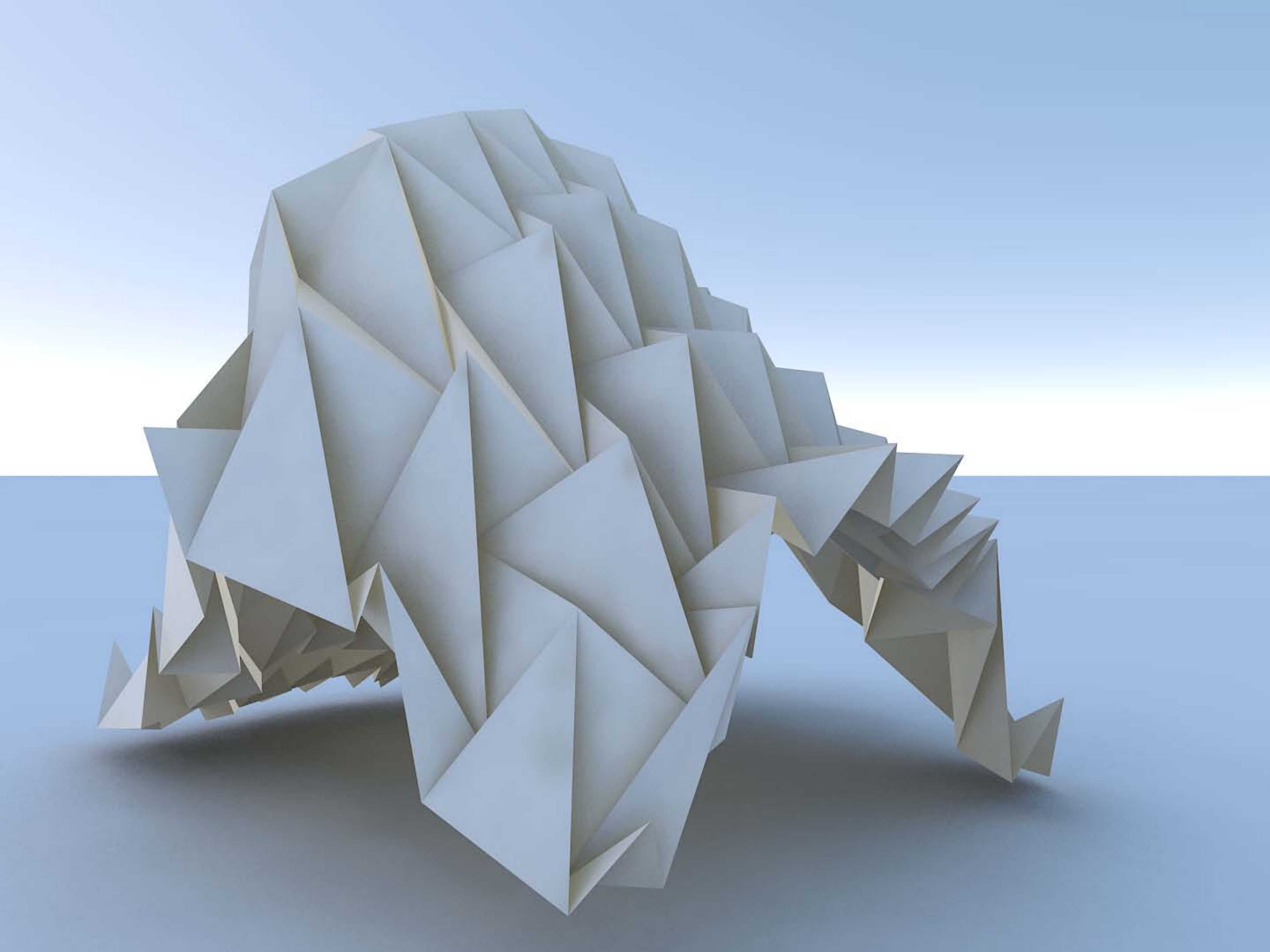



Kuribayashi & You 2006

# Waterbomb Pattern Generalized







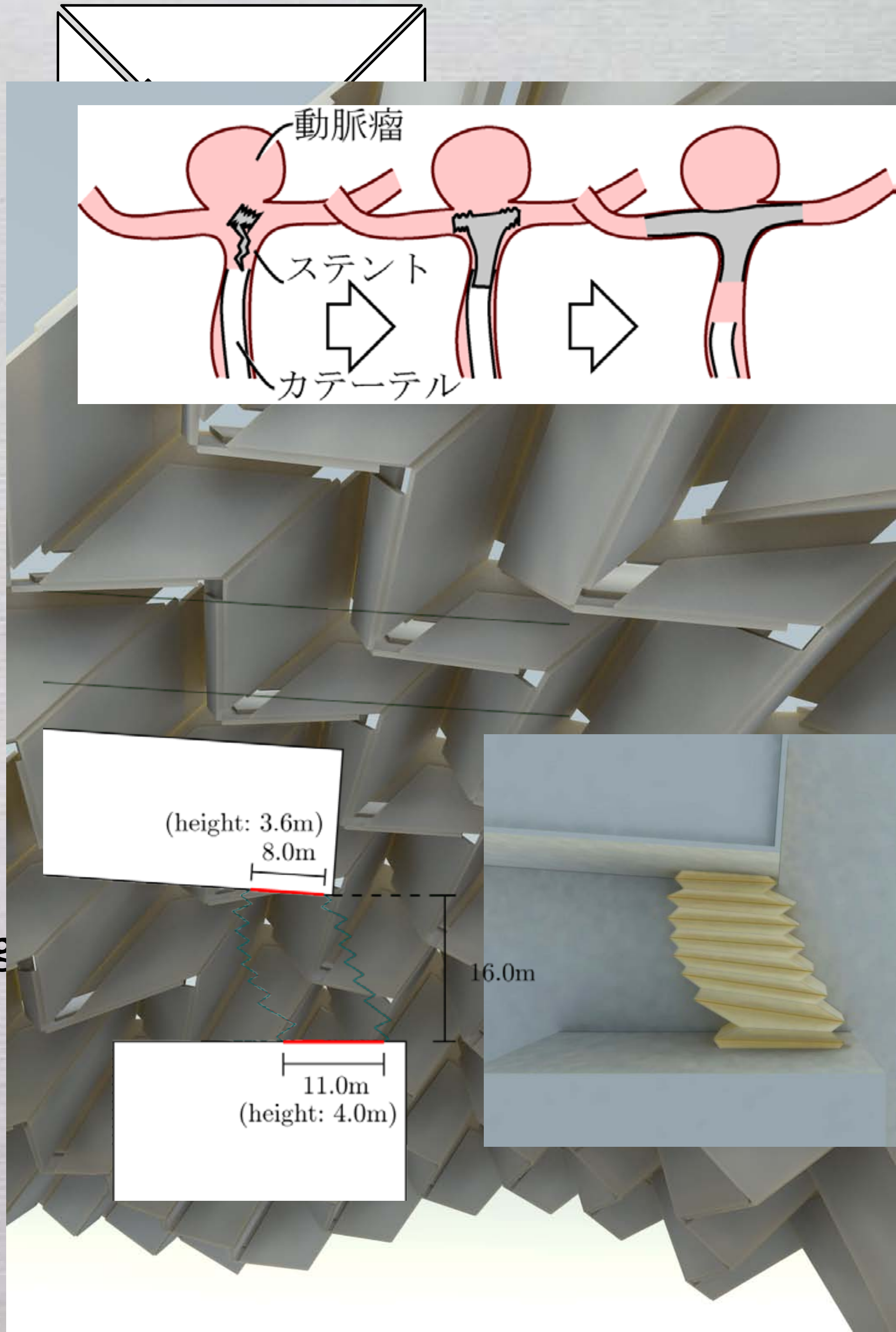
# 3

# Rigid Origami


- Tachi T.: "Rigid-Foldable Thick Origami", in Origami5, to appear.
- Tachi T.: "Freeform Rigid-Foldable Structure using Bidirectionally Flat-Foldable Planar Quadrilateral Mesh", Advances in Architectural Geometry 2010, pp. 87--102, 2010.
- Miura K. and Tachi T.: "Synthesis of Rigid-Foldable Cylindrical Polyhedra," Journal of ISIS-Symmetry, Special Issues for the Festival-Congress Gmuend, Austria, August 23-28, pp. 204-213, 2010.
- Tachi T.: "One-DOF Cylindrical Deployable Structures with Rigid Quadrilateral Panels," in Proceedings of the IASS Symposium 2009, pp. 2295-2306, Valencia, Spain, September 28- October 2, 2009.
- Tachi T.: "Generalization of Rigid-Foldable Quadrilateral-Mesh Origami," Journal of the International Association for Shell and Spatial Structures (IASS), 50(3), pp. 173–179, December 2009.
- Tachi T.: "Simulation of Rigid Origami ," in Origami4, pp. 175-187, 2009.

# Rigid Origami?

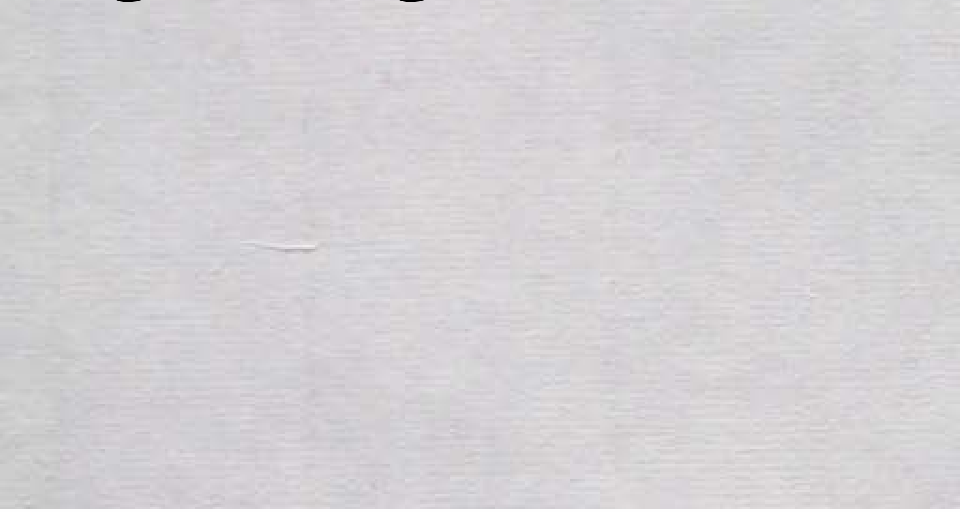
- Rigid Origami is
  - Plates and Hinges model for origami
- Characteristics
  - Panels do not deform
    - Do not use Elasticity
  - synchronized motion
    - Especially nice if One-DOF
  - watertight cover for a space
- Applicable for
  - self deployable micro mechanism
  - large scale objects under gravity using thick panels

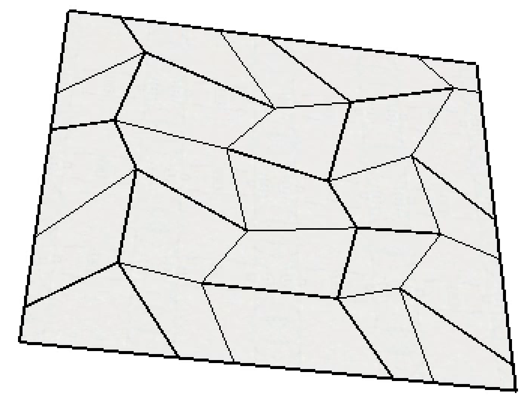


# Study Objectives

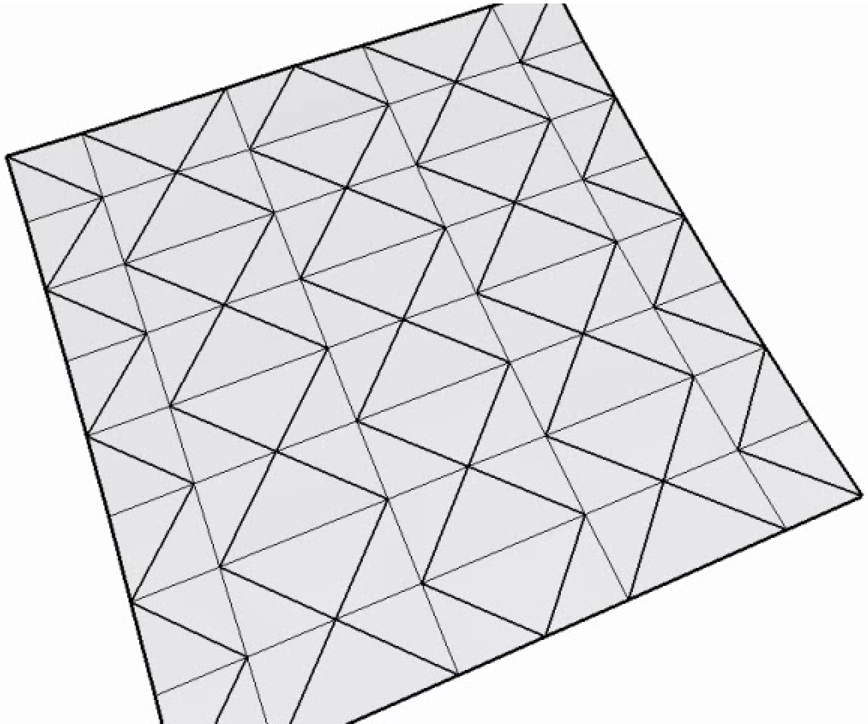
1. Generalize rigid foldable structures to freeform
    1. Generic triangular-mesh based design
        - multi-DOF
        - statically determinate
    2. Singular quadrilateral-mesh based design
        - one-DOF
        - redundant contraints
2. Generalize rigid foldable structures to cylinders and more

# Examples of Rigid Origami

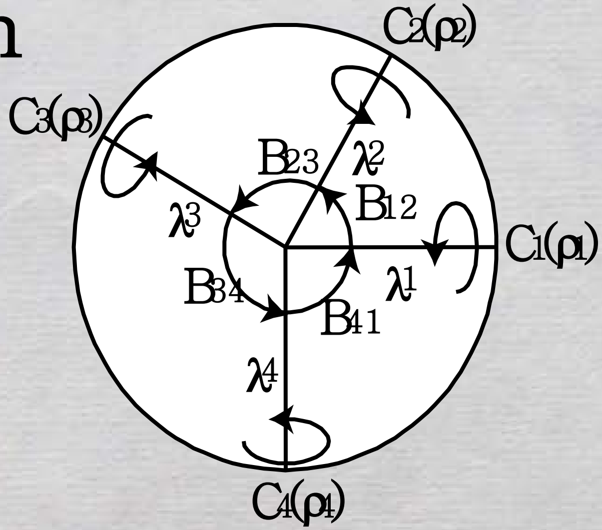








# Basics of Rigid Origami Angular Representation




$$\chi_1 = \mathbf{C}_1\mathbf{B}_{12}$$
$$\chi_2 = \mathbf{C}_2\mathbf{B}_{23}$$
$$\chi_3 = \mathbf{C}_3\mathbf{B}_{34}$$
$$\chi_4 = \mathbf{C}_4\mathbf{B}_{41}$$

- Constraints
  - [Kawasaki 87] [belcastro and Hull 02]

$$\chi_1 \cdots \chi_{n-1}\chi_n = \mathbf{I}$$

  - 3 equations per interior vertex
- $V_{in}$ interior vert + $E_{in}$ foldline model:
  - constraints:

$$\underbrace{[\mathbf{C}]}_{3V_{in}\times E_{in}\ \text{matrix}} \quad \dot{\boldsymbol{\rho}} = \mathbf{0}$$

Generic case:
DOF = $E_{in}$ $-3V_{in}$

$$\dot{\boldsymbol{\rho}} = \left[\mathbf{I}_N - \mathbf{C}^+\mathbf{C}\right]\dot{\boldsymbol{\rho}}_0$$

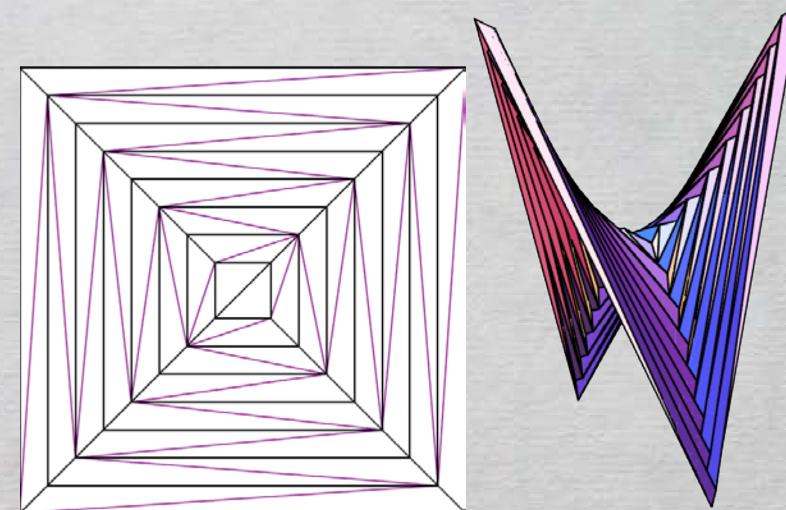
$$\left(\begin{array}{l}\text{where } \mathbf{C}^+ \text{ is the} \\ \text{pseudo-inverse of } \mathbf{C}\end{array}\right)$$

# DOF in Generic Triangular Mesh

Euler's: $(V_{in}+E_{out})-(E_{out}+E_{in})+F=1$

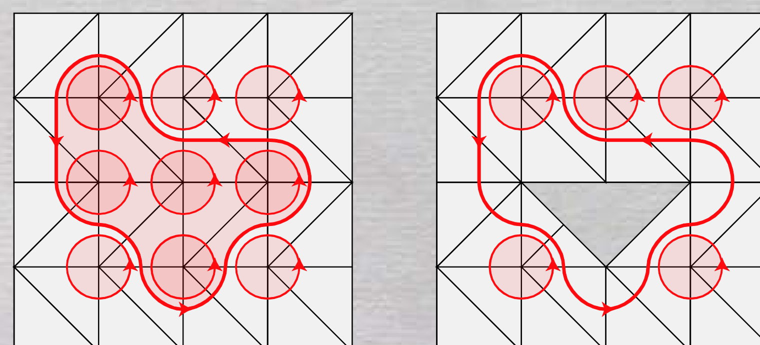
Triangle : $3F=2E_{out} + E_{in}$

Mechanism: $DOF = E_{in} -3V_{in}$



Disk with $E_{out}$ outer edges

$$DOF = E_{out} -3$$

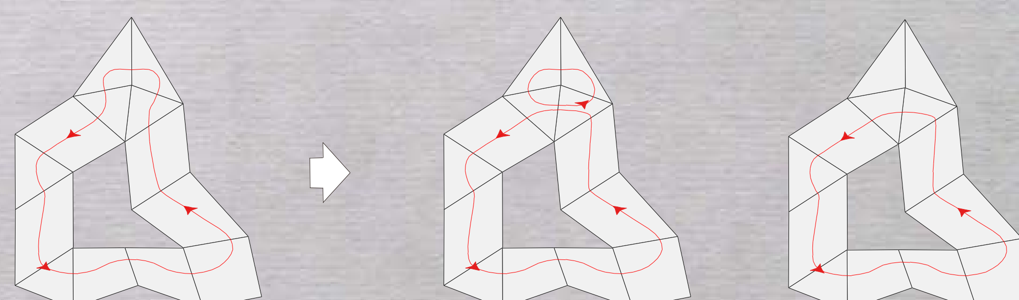


with H generic holes

$DOF = E_{out} -3 -3H$

$(V_{in}+E_{out})-(E_{out}+E_{in})+F=1-H$

$DOF = E_{in} -3V_{in}-6H$



# Hexagonal Tripod Shell

Hexagonal boundary:

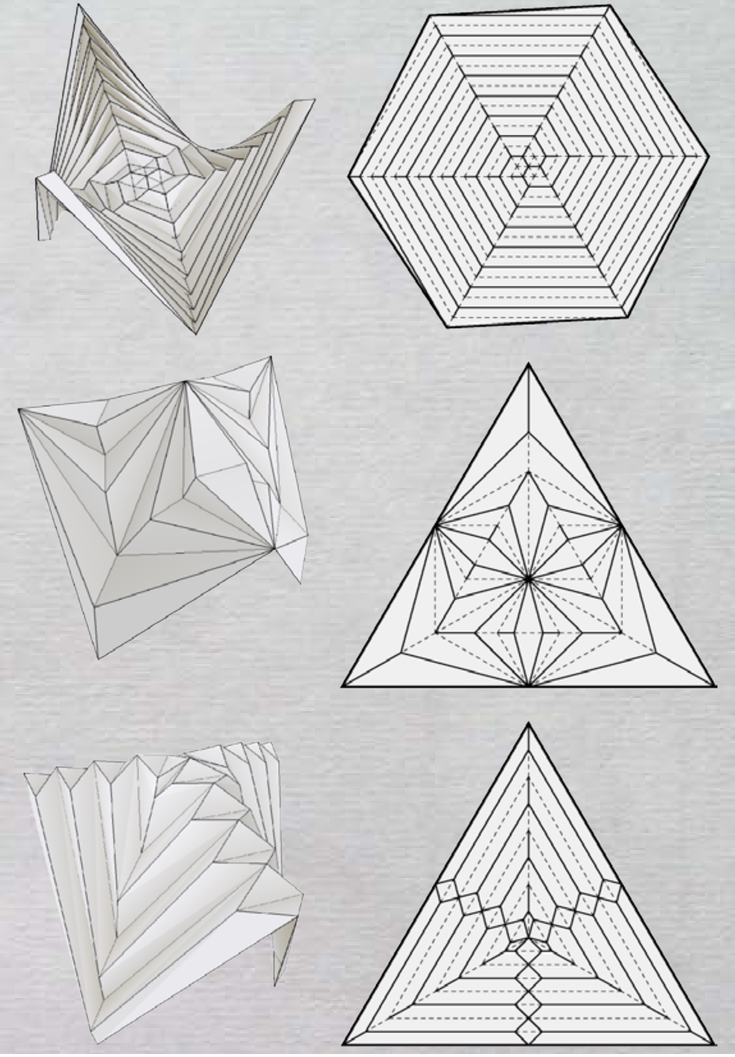
$E_{out} = 6$

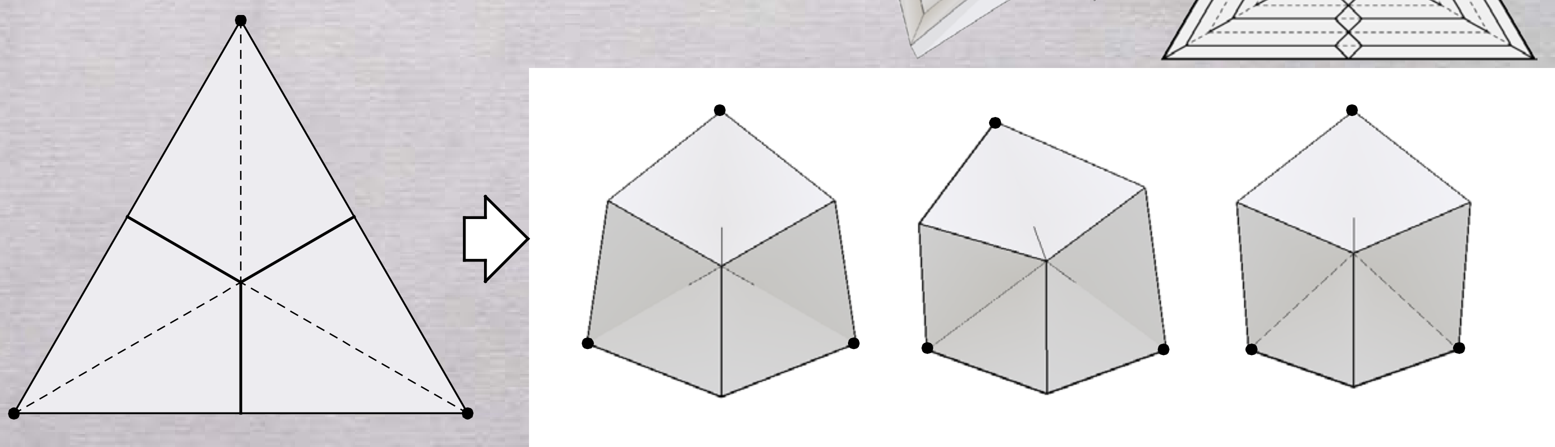
∴ DOF = 6 - 3 =3

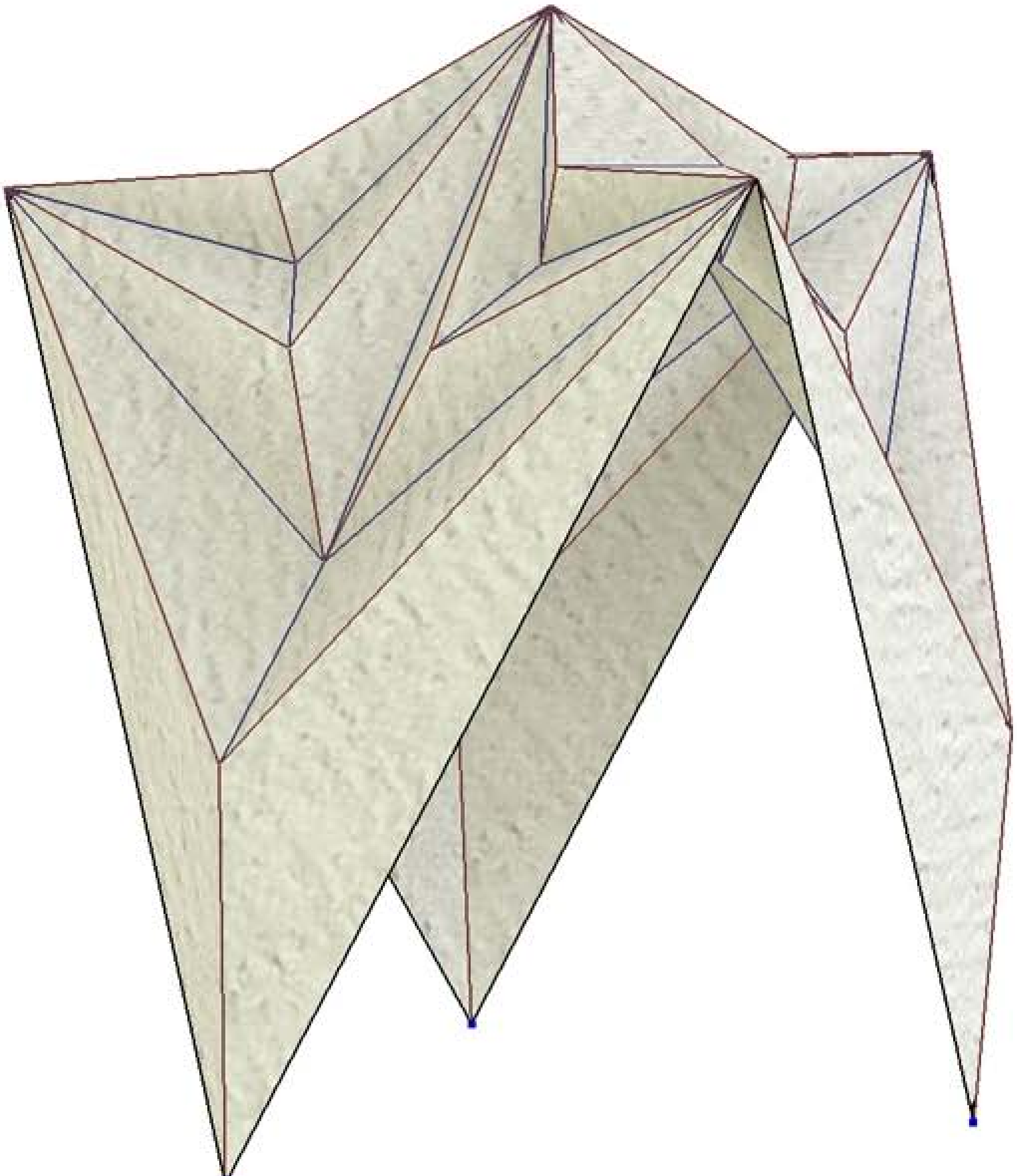
+ rigid DOF = 6

3 pin joints (x,y,z):

∴ 3x3 = 9 constraints

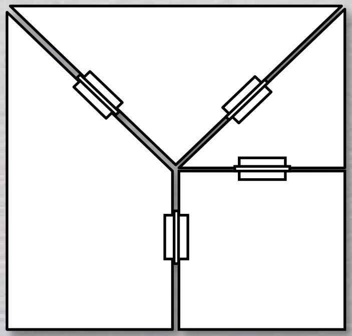




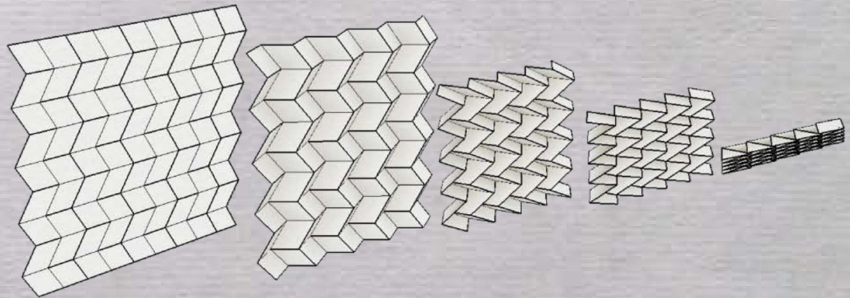


# Generalize Rigid-Foldable Planar Quad-Mesh

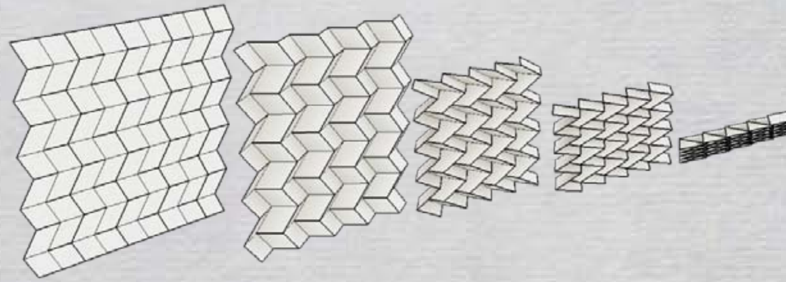
- One-DOF
  - Every vertex transforms in the same way
  - **Controllable with single actuator**
- Redundant
  - Rigid Origami in General
    - DOF = N – 3M
    - N: num of foldlines
    - M: num of inner verts
  - nxn array N=2n(n-1), M=(n-1)2
    -> DOF=-(n-2)2+1
    -> n>2, then overconstrained if not singular
  - Rank of Constraint Matrix is N-1
    - Singular Constraints
  - **Robust structure**
  - **Improved Designability**

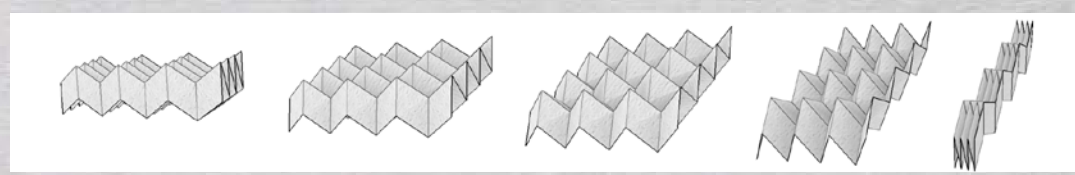


N=4, M=1
DOF = 1

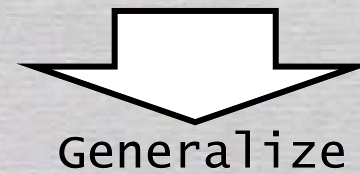


# Idea: Generalize Regular pattern





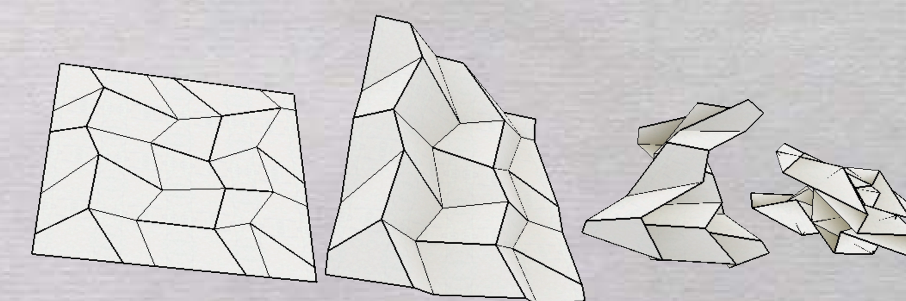
- Original
  - Miura-ori
  - Eggbox pattern



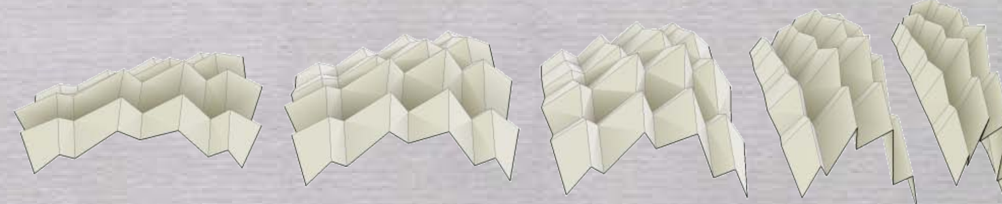

- Generalization

  To:

  Non Symmetric forms

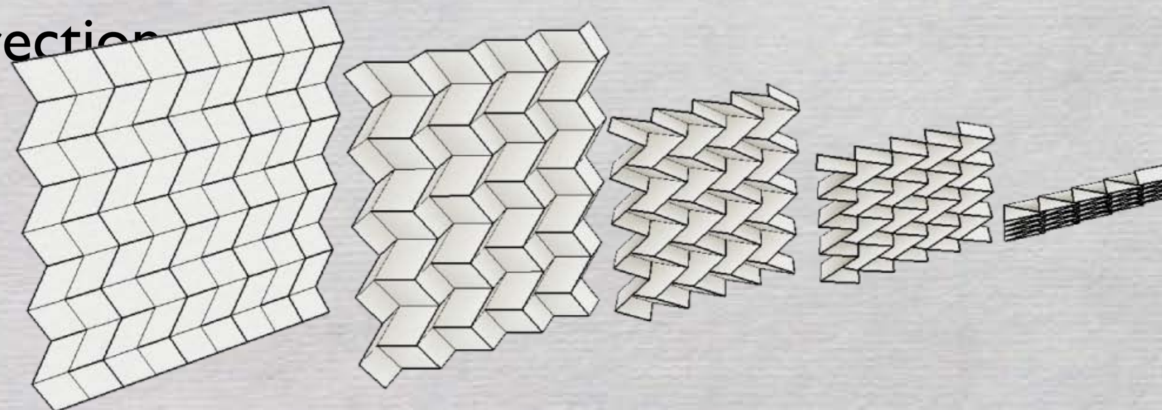


(Do not break rigid foldability)

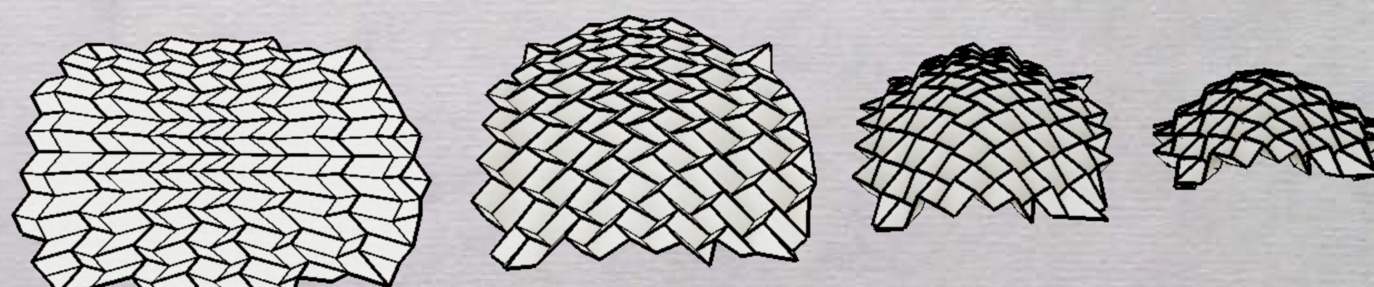


# Flat-Foldable Quadrivalent Origami MiuraOri Vertex

- one-DOF structure
  - x,y in the same direction



- Miura-ori
- Variation of Miura-ori

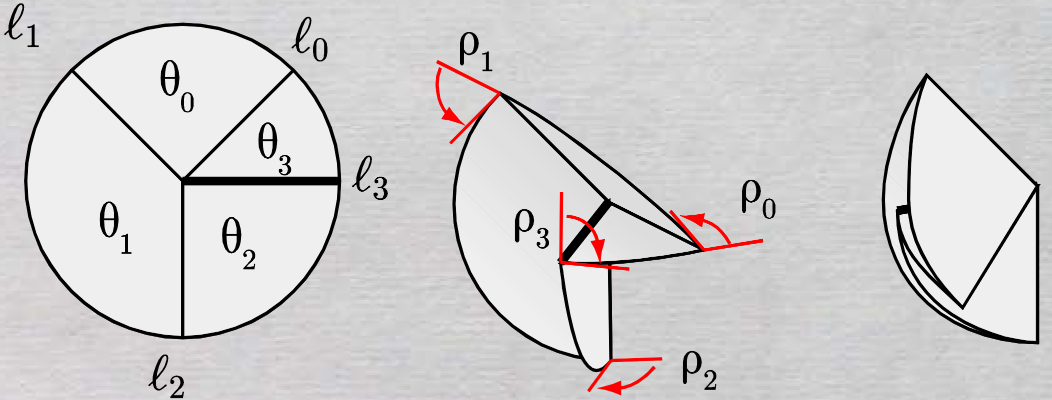


# Flat-Foldable Quadrivalent Origami MiuraOri Vertex

- Intrinsic Measure:

$$\theta_0 = \pi - \theta_2$$

$$\theta_1 = \pi - \theta_3$$



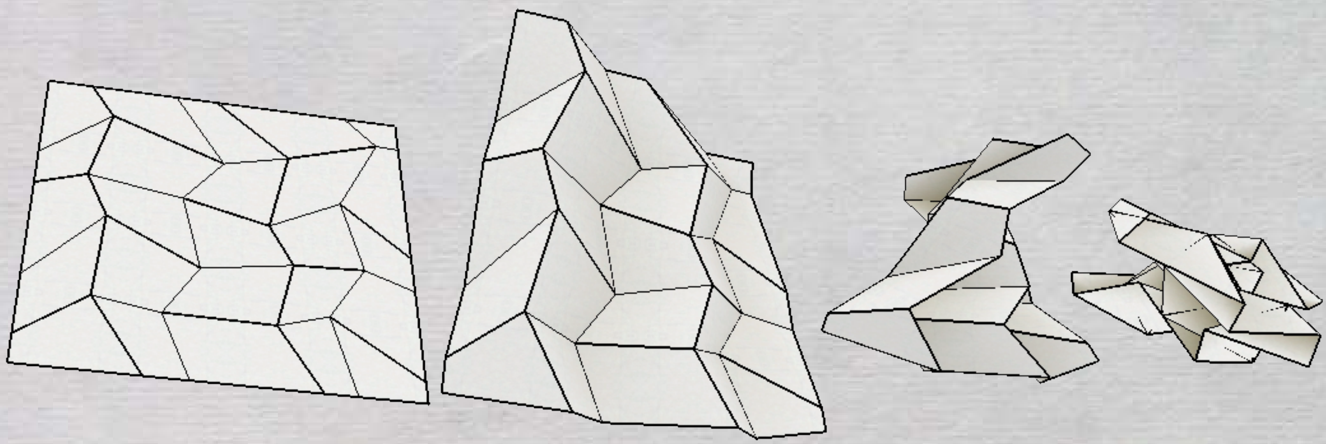

- Folding Motion
  - Opposite fold angles are equal
  - Two pairs of folding motions are linearly related.

$$\rho_1 = -\rho_3$$

$$\rho_0 = \rho_2$$

$$\tan\frac{\rho_0}{2} = \sqrt{\frac{1+\cos(\theta_0 - \theta_1)}{1+\cos(\theta_0 + \theta_1)}}\tan\frac{\rho_1}{2}$$

# Flat-Foldable Quadrivalent Origami
# MiuraOri Vertex



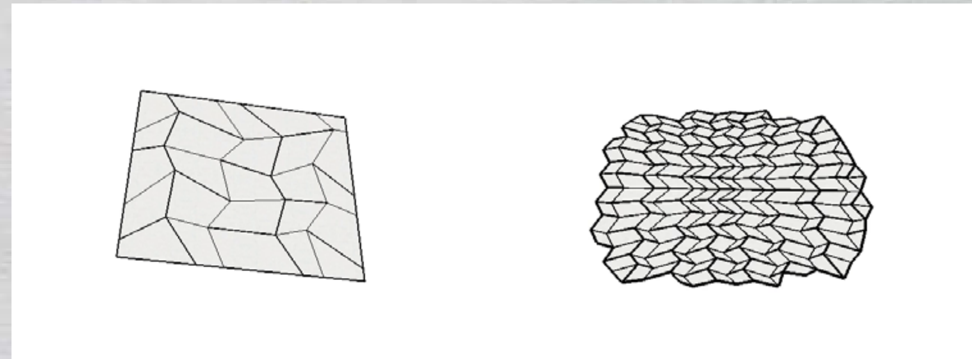
$$\begin{bmatrix} \tan\frac{\rho_1(t)}{2} \\ \tan\frac{\rho_2(t)}{2} \\ \vdots \\ \tan\frac{\rho_N(t)}{2} \end{bmatrix} = \lambda(t) \begin{bmatrix} \tan\frac{\rho_1(t_0)}{2} \\ \tan\frac{\rho_2(t_0)}{2} \\ \vdots \\ \tan\frac{\rho_N(t_0)}{2} \end{bmatrix}$$

$$\rho_1 = -\rho_3$$

$$\rho_0 = \rho_2$$

$$\tan\frac{\rho_0}{2} = \sqrt{\frac{1+\cos(\theta_0 - \theta_1)}{1+\cos(\theta_0 + \theta_1)}} \tan\frac{\rho_1}{2}$$

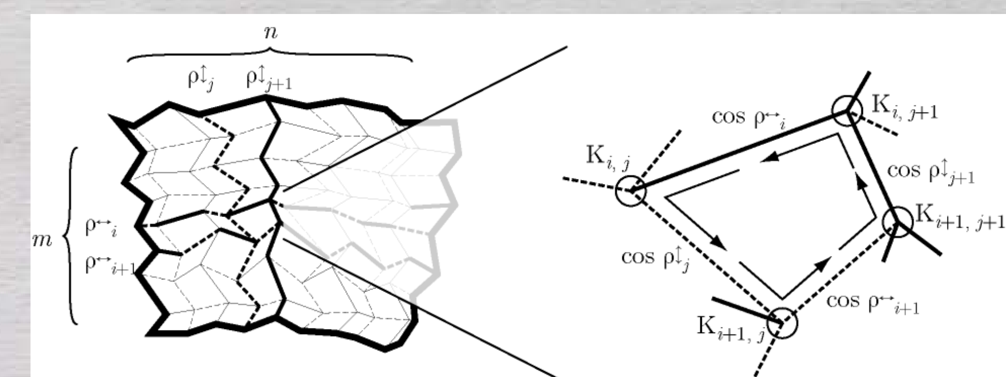
# Get One State and Get Continuous Transformation



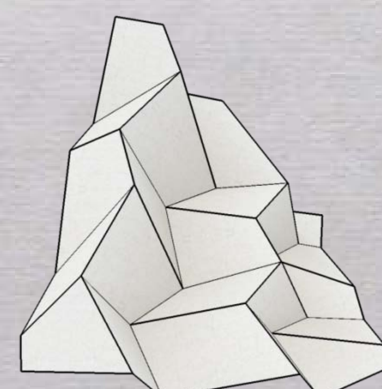
Finite Foldability: Existence of
 Folding Motion $\Leftrightarrow$

There is one static state with

- Developability
- Flat-foldability
- Planarity of Panels



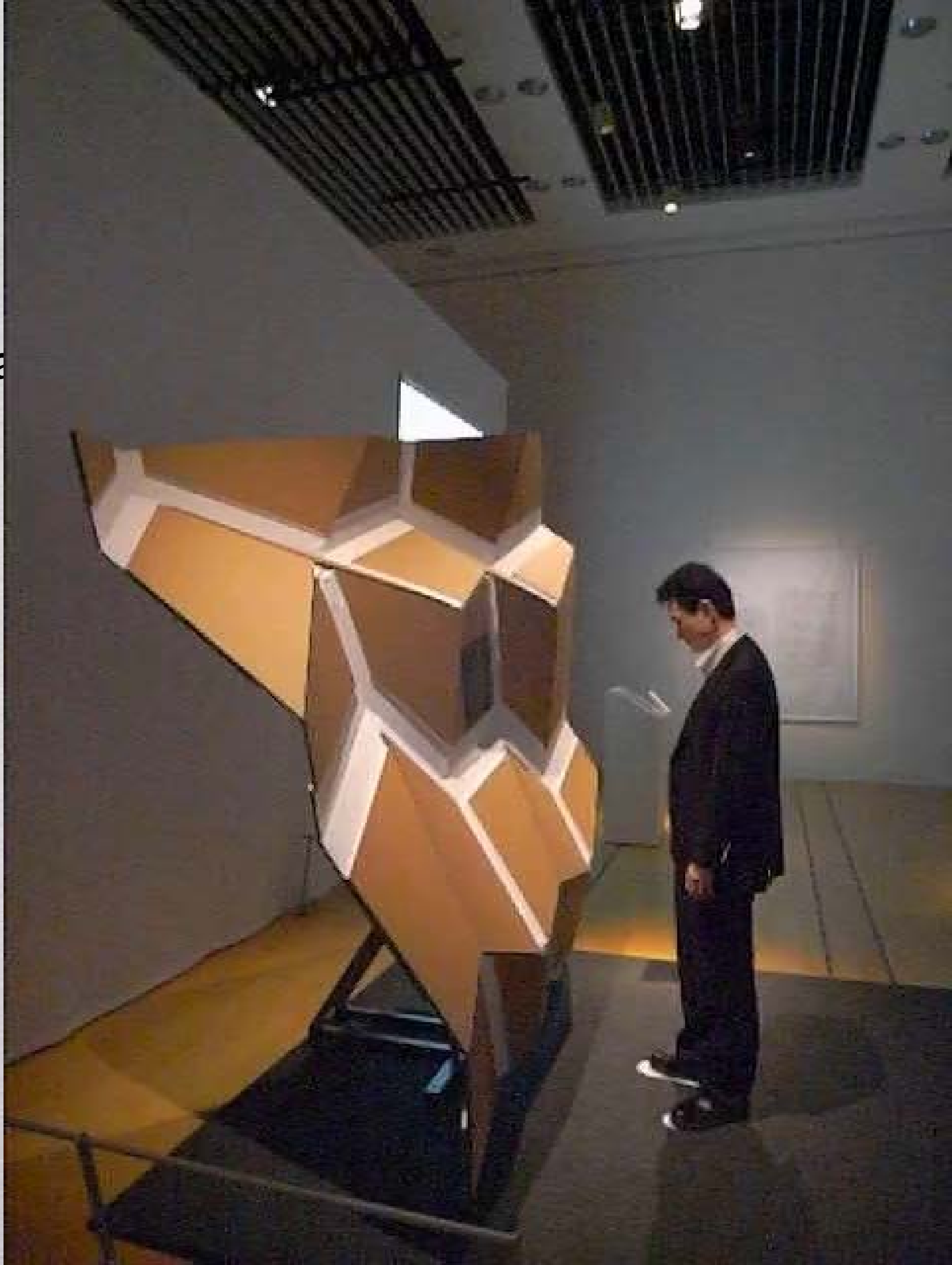

$$\begin{bmatrix} \tan\dfrac{\rho_1(t)}{2} \\ \tan\dfrac{\rho_2(t)}{2} \\ \vdots \\ \tan\dfrac{\rho_N(t)}{2} \end{bmatrix} = \lambda(t) \begin{bmatrix} \tan\dfrac{\rho_1(t_0)}{2} \\ \tan\dfrac{\rho_2(t_0)}{2} \\ \vdots \\ \tan\dfrac{\rho_N(t_0)}{2} \end{bmatrix}$$

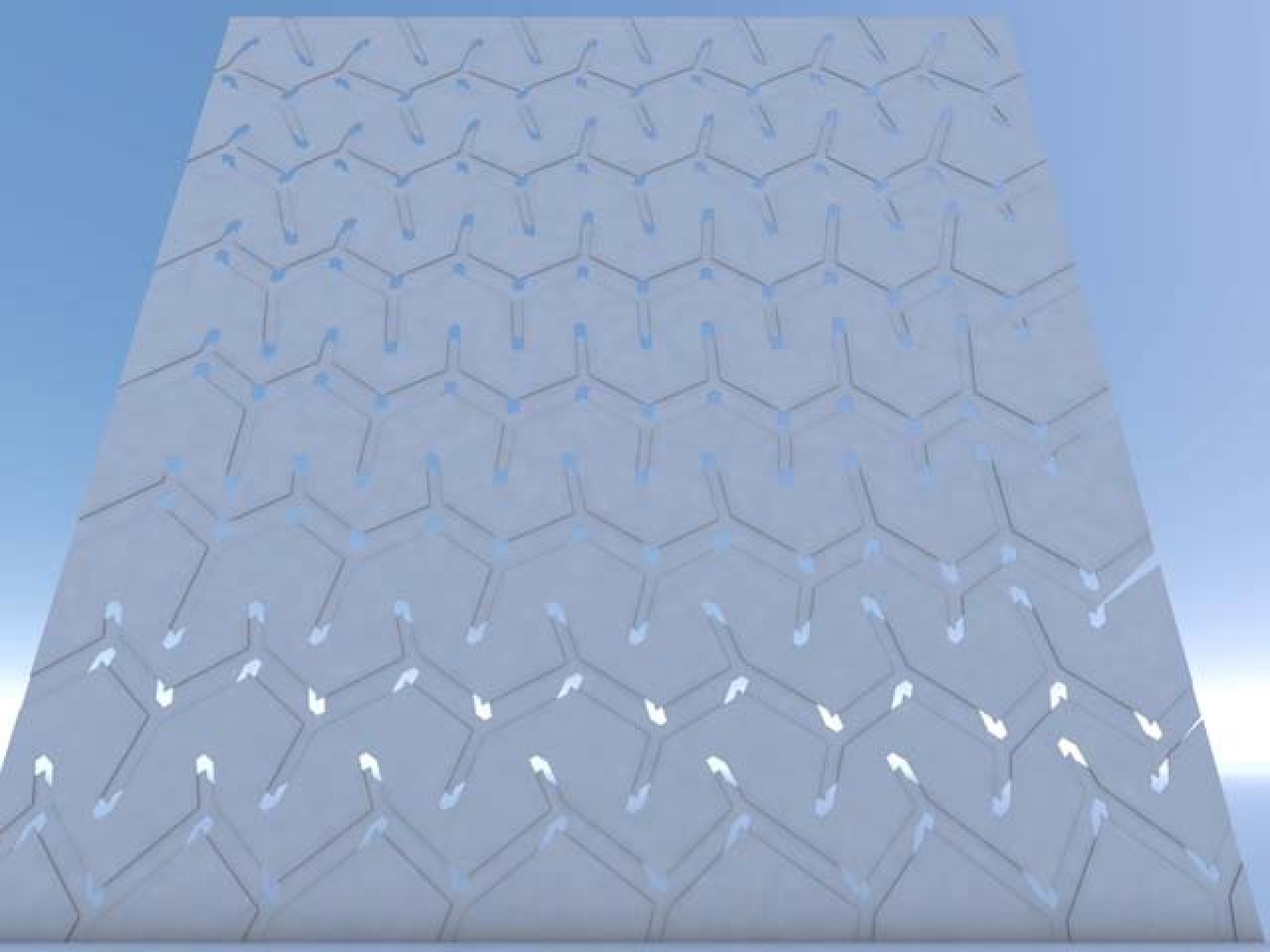


# Built Design

- Material
  - 10mm Structural Cardboa
    (double wall)
  - Cloth
- Size
  - 2.5m x 2.5m

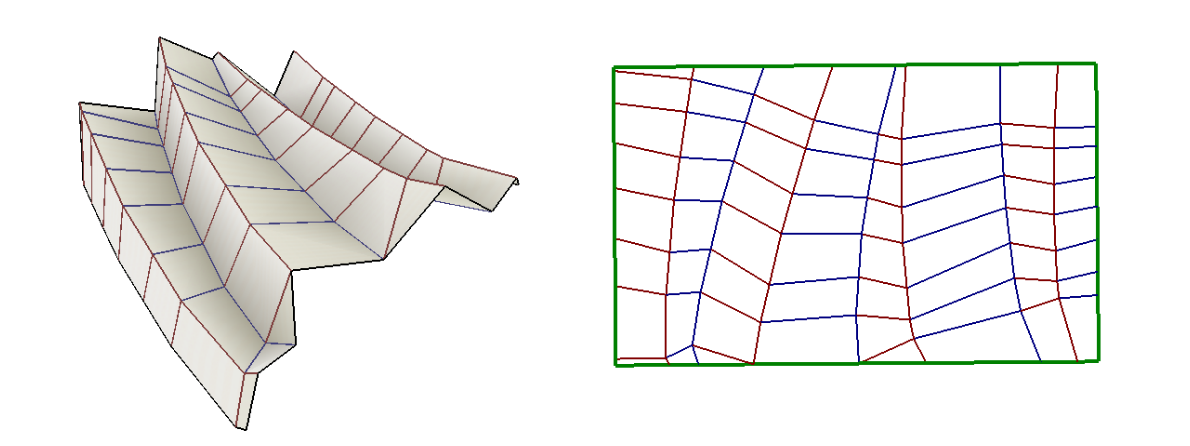
- exhibited at NTT ICC

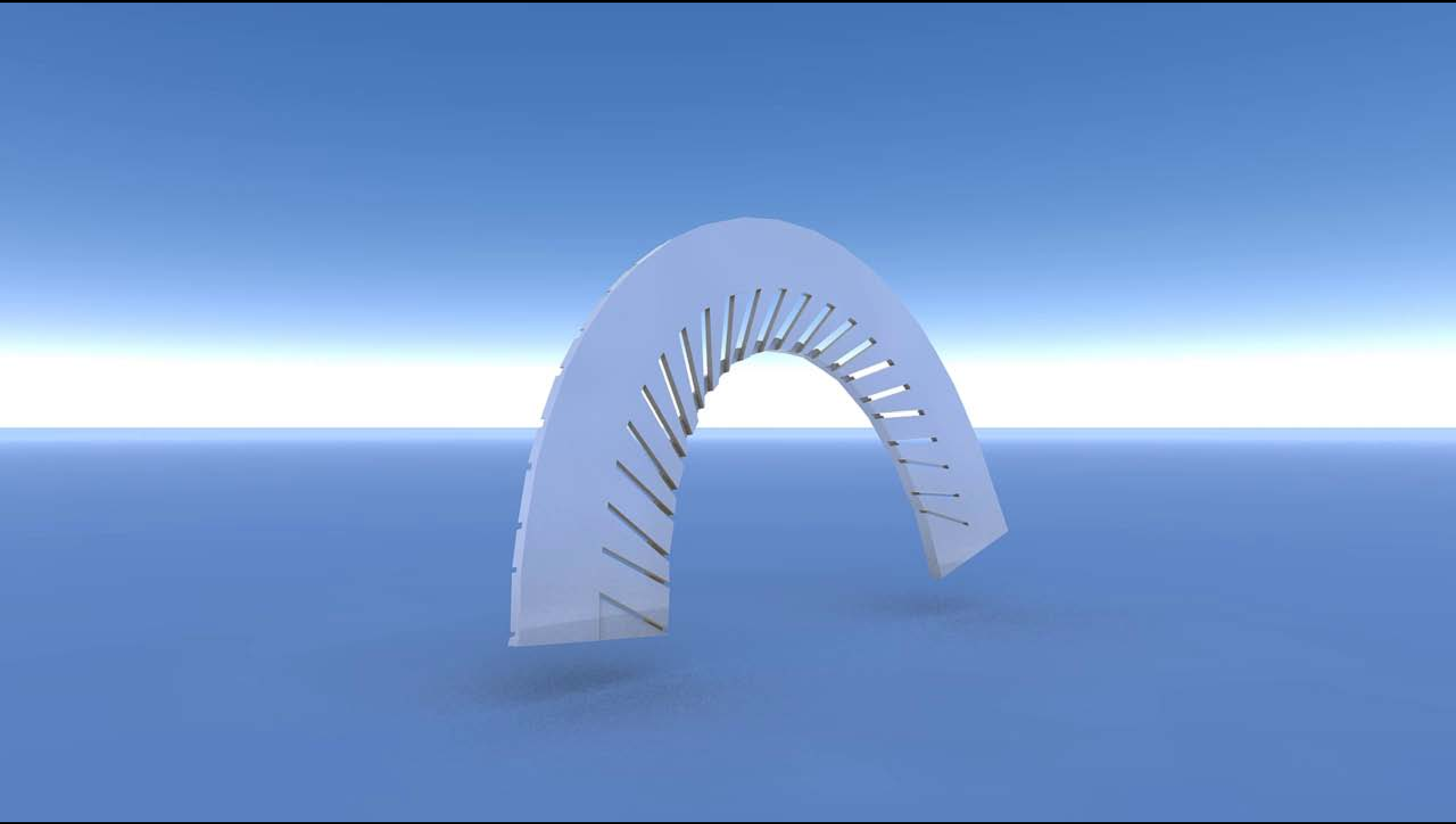




# Rigid Foldable Curved Folding

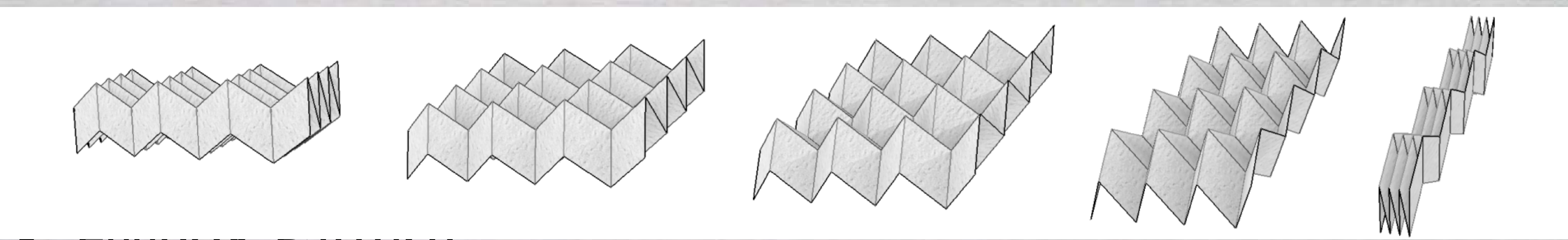
- Curved folding is rationalized by Planar Quad Mesh
- Rigid Foldable Curved Folding
  =
  Curved folding without ruling sliding



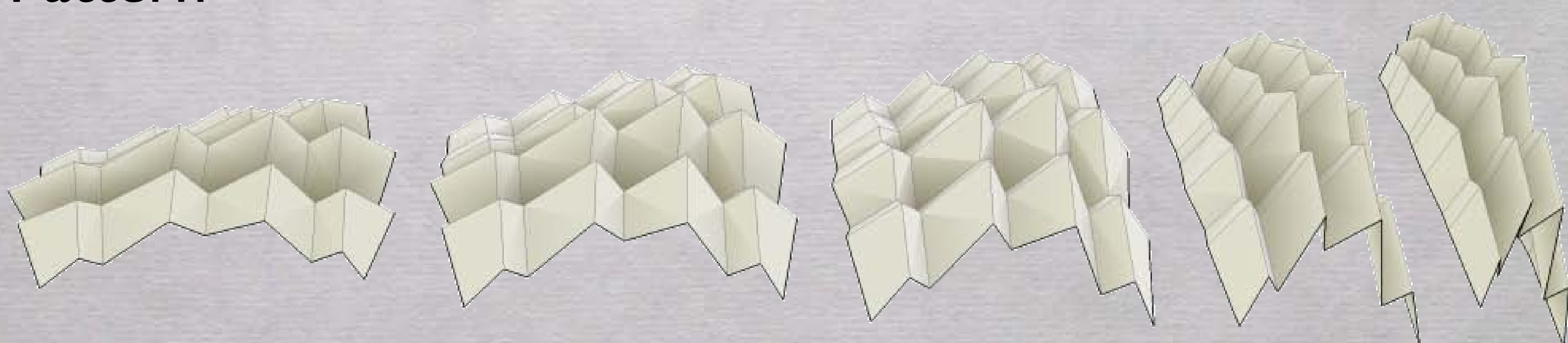


# Discrete Voss Surface Eggbox-Vertex

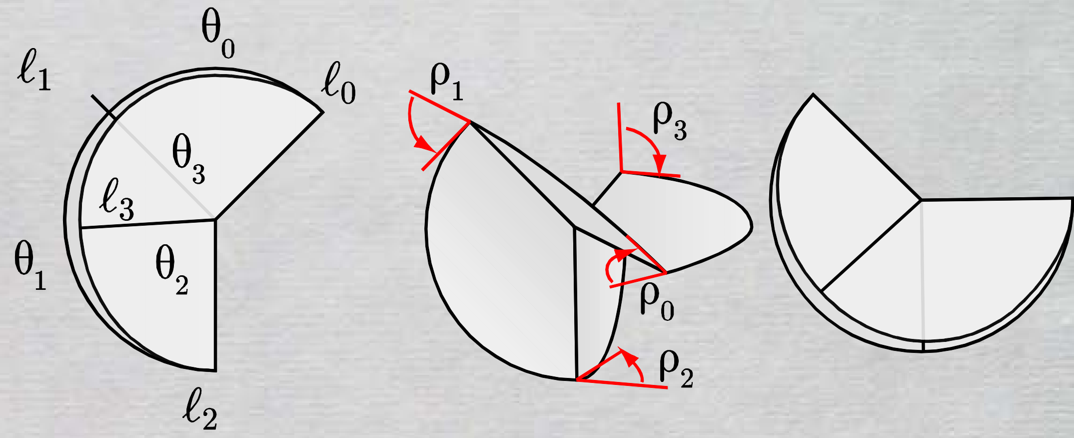
- one-DOF structure
  - Bidirectionally Flat-Foldable



- Eggbox-Pattern
- Variation of Eggbox Pattern



# Discrete Voss Surface
# Eggbox-Vertex




- Intrinsic Measure:

$$\theta_0 = \theta_2$$

$$\theta_1 = \theta_3$$

- Folding Motion
  - Opposite fold angles are equal
  - Two pairs of folding motions are linearly related. [SCHIEF et.al. 2007]

Complementary Folding Angle

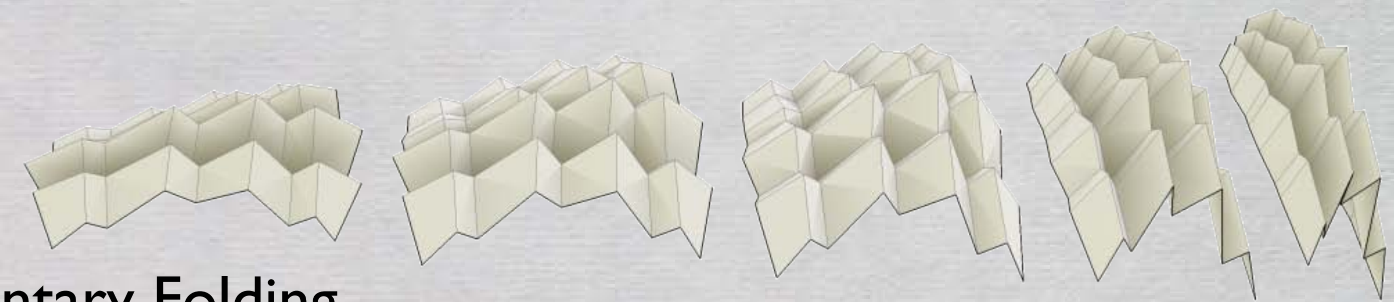
$$\rho_1 = \rho_3 \quad = \pi - \rho_1' = \pi - \rho_3'$$

$$\rho_0 = \rho_2$$

$$\tan\frac{\rho_0}{2} = \sqrt{\frac{1+\cos(\theta_0 - \theta_1)}{1+\cos(\theta_0 + \theta_1)}}\cot\frac{\rho_1}{2}$$

$$= \sqrt{\frac{1+\cos(\theta_0 - \theta_1)}{1+\cos(\theta_0 + \theta_1)}}\tan\frac{\rho_1'}{2}$$

# Eggbox: Discrete Voss Surface



- Use Complementary Folding Angle for "Complementary Foldline"

Complementary Folding Angle

$$\rho_1 = \rho_3 \quad = \pi - \rho_1' = \pi - \rho_3'$$

$$\rho_0 = \rho_2$$

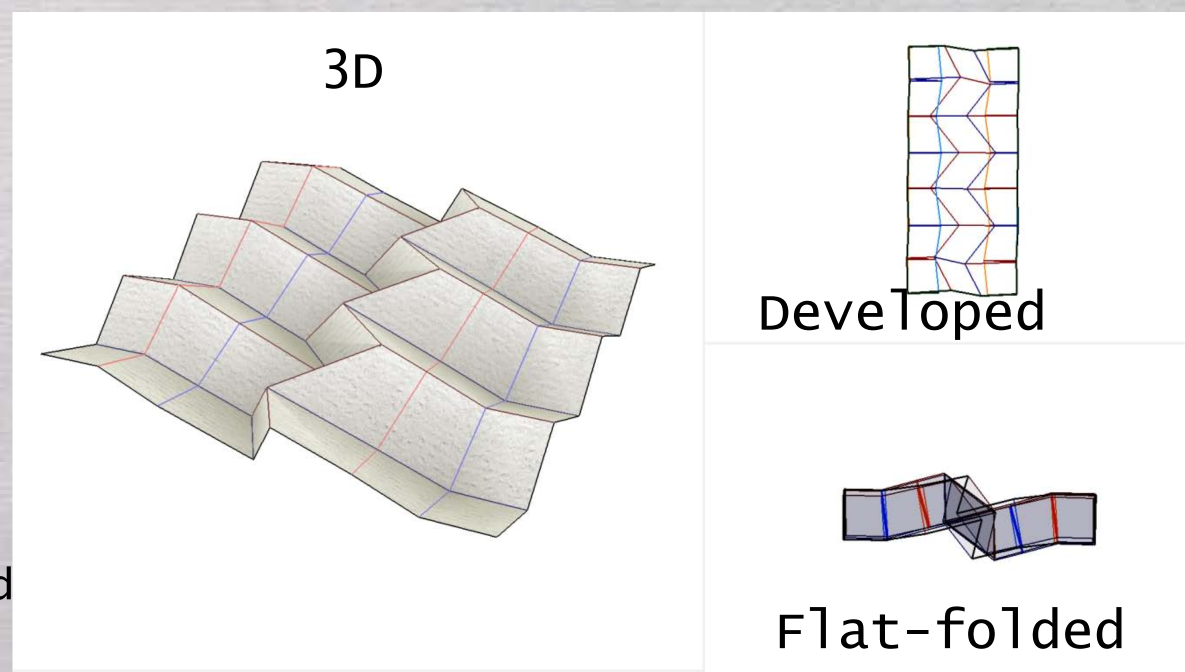
$$\tan\frac{\rho_0}{2} = \sqrt{\frac{1+\cos(\theta_0-\theta_1)}{1+\cos(\theta_0+\theta_1)}}\cot\frac{\rho_1}{2} = \sqrt{\frac{1+\cos(\theta_0-\theta_1)}{1+\cos(\theta_0+\theta_1)}}\tan\frac{\rho_1'}{2}$$

$$\begin{bmatrix} \tan\frac{\rho_0(t)}{2} \\ \tan\frac{\rho_1(t)}{2} \\ \vdots \\ \tan\frac{\rho_N(t)}{2} \end{bmatrix} = \lambda(t)\begin{bmatrix} \tan\frac{\rho_0(t_0)}{2} \\ \tan\frac{\rho_1(t_0)}{2} \\ \vdots \\ \tan\frac{\rho_N(t_0)}{2} \end{bmatrix}$$

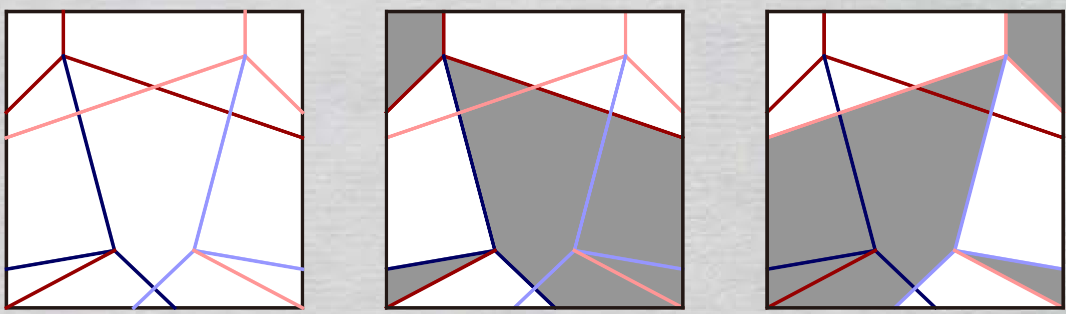
# Hybrid Surface: BiDirectionally Flat-Foldable PQ Mesh

- use 4 types of foldlines
  - mountain fold
    - 0° -> -180°
  - valley fold
    - 0° -> 180°
  - complementary mountain fold
    - -180° -> 0°
  - complementary valley fold
    - 180° -> 0°

“developed” state    flat-folded state



# Developability and Flat-Foldability



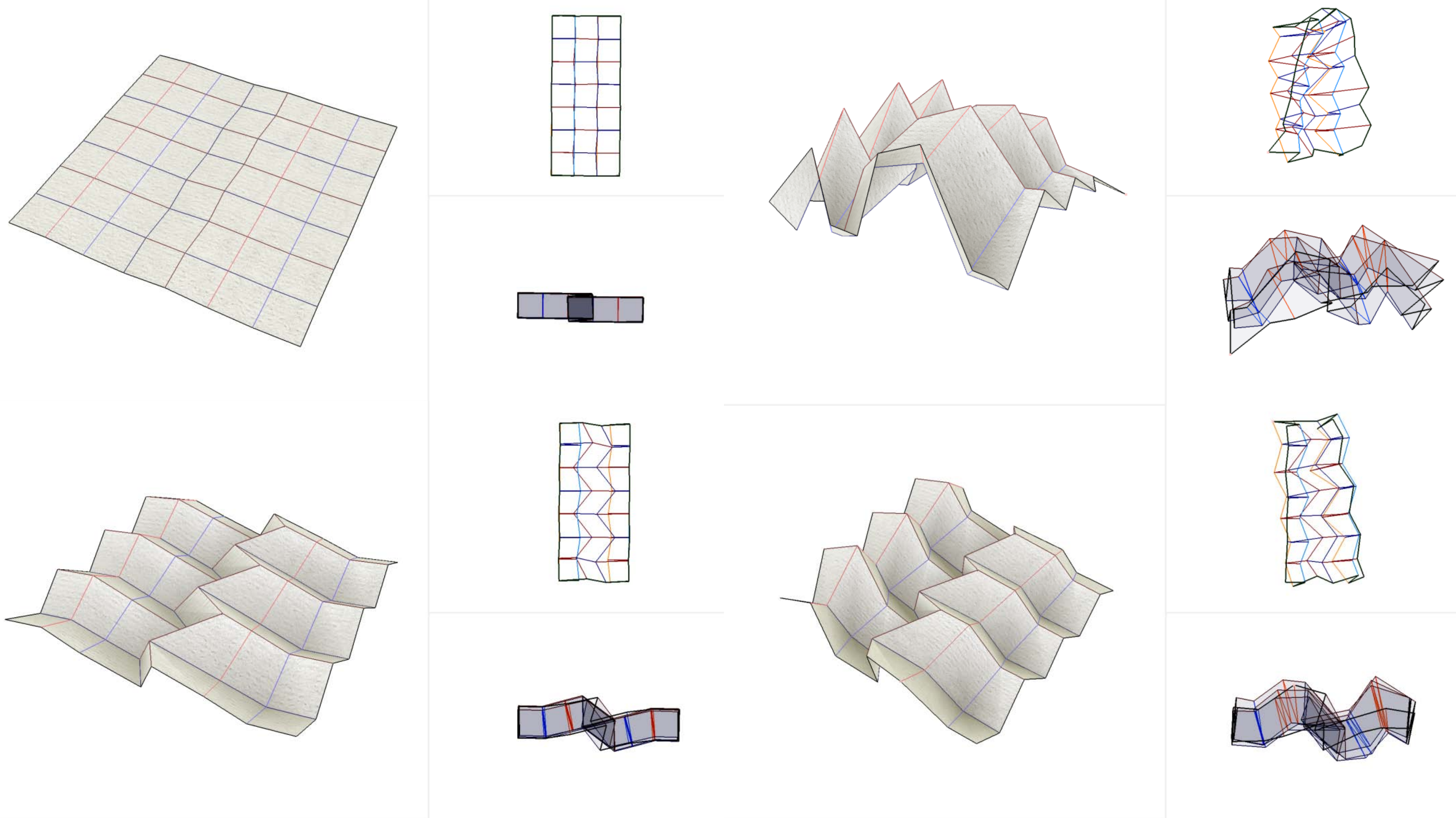

- Developed State:
  - Every edge has fold angle complementary fold angle be 0°

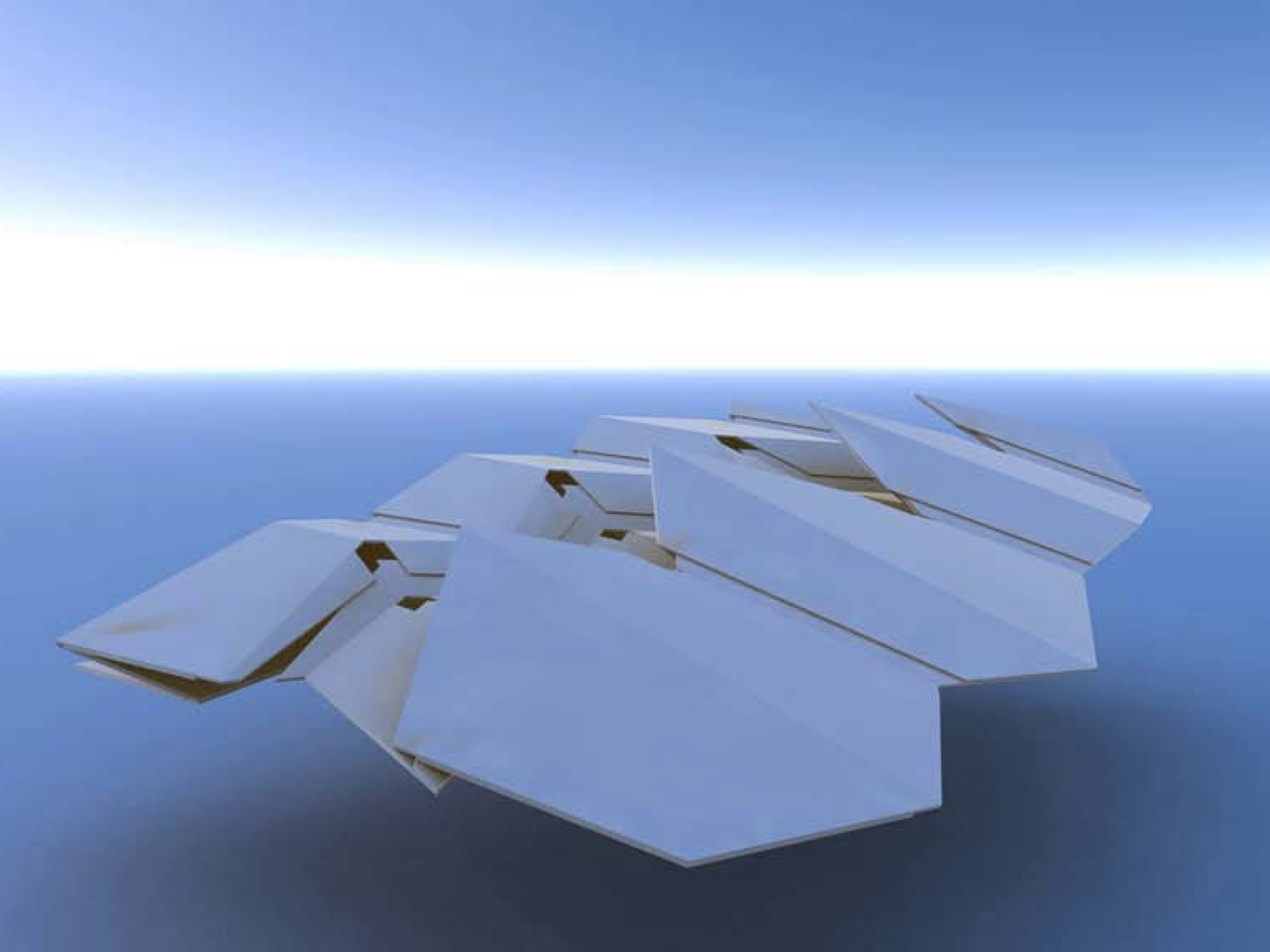
$$\begin{cases} \sum_{i=0}^{3} \sigma^{dev}(i)\theta_i = 0 & \cdots 4CF \quad or \quad 2F+2CF \\ 2\pi - \sum_{i=0}^{3} \theta_i = 0 & \cdots \quad 4F \end{cases}$$

- Flat-folded State:
  - Every edge has fold angle complementary fold angle to be ±180°

$$\begin{cases} \sum_{i=0}^{3} \sigma^{ff}(i)\theta_i = 0 & \cdots 4F \quad or \quad 2F+2CF \\ 2\pi - \sum_{i=0}^{3} \theta_i = 0 & \cdots \quad 4CF \end{cases}$$

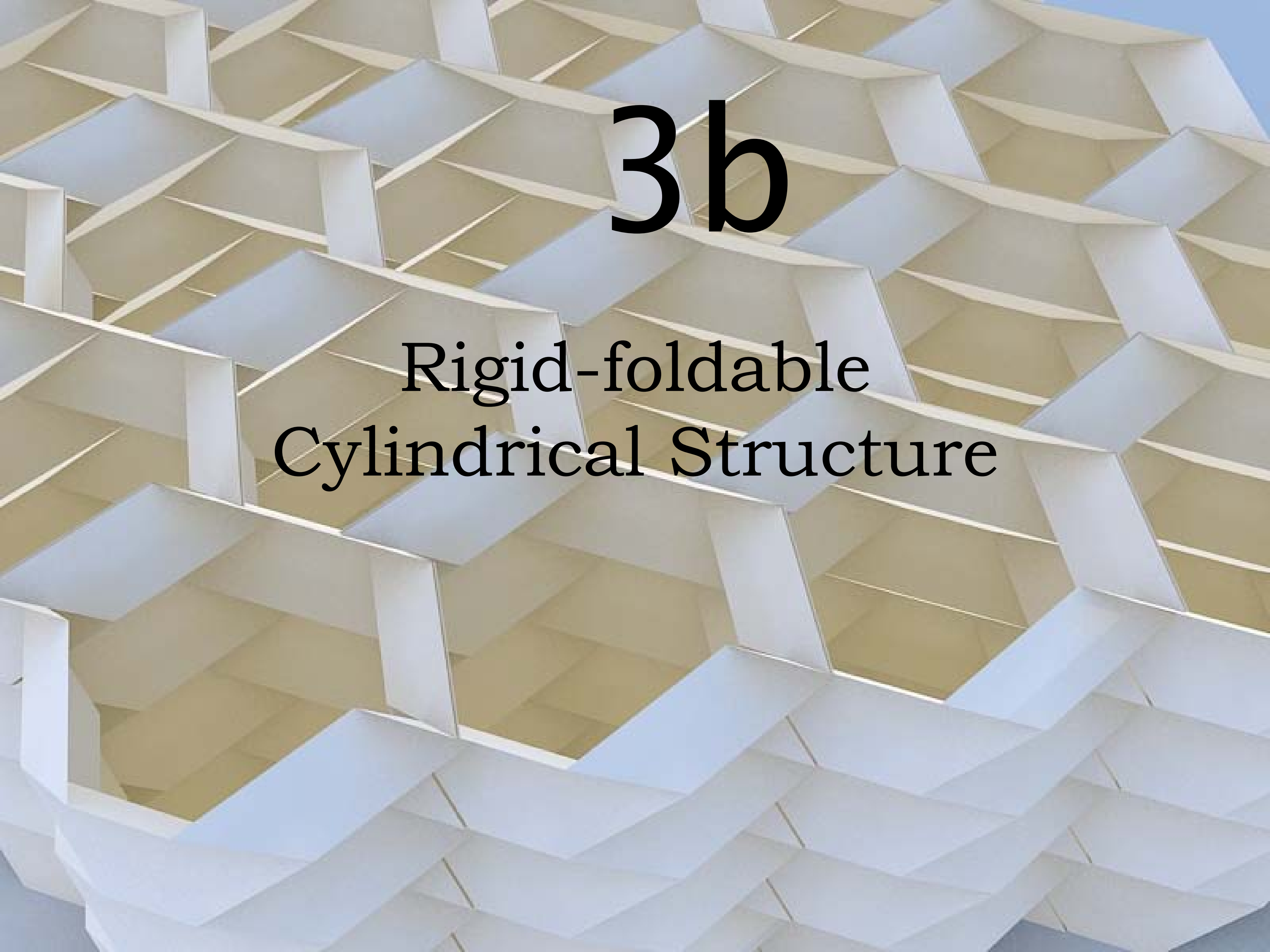
# Hybrid Surface:
# BiDirectionally Flat-Foldable PQ Mesh





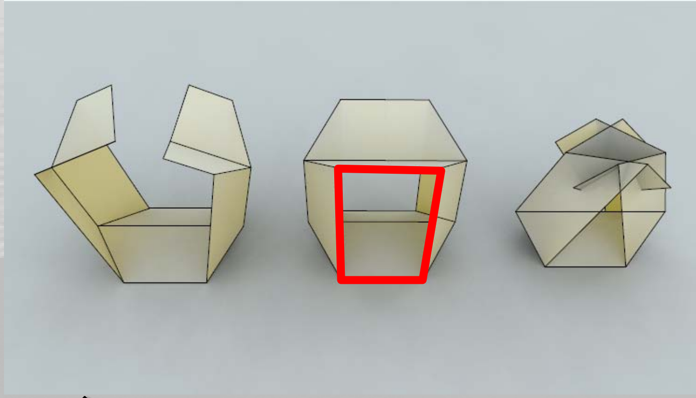
# 3b

## Rigid-foldable Cylindrical Structure

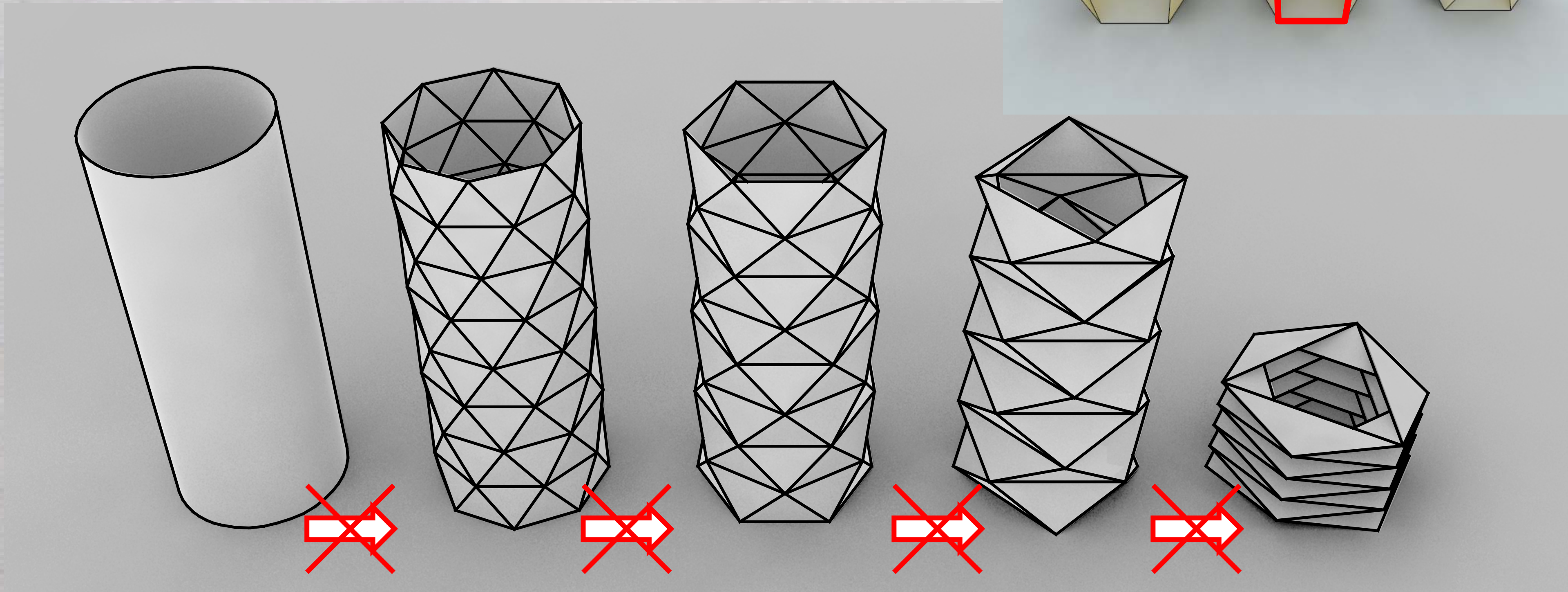


# Topologically Extend Rigid Origami

- Generalize to the cylindrical, or higher genus rigid-foldable polyhedron.



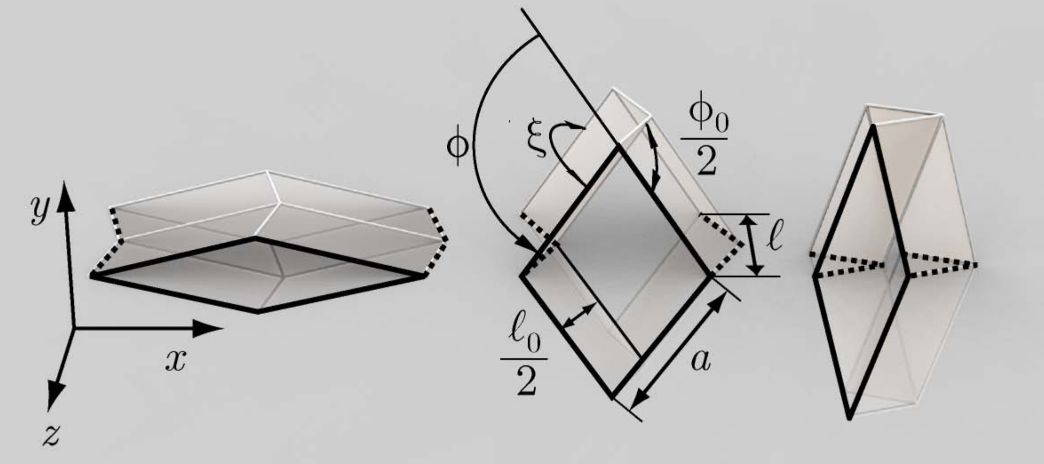
- But it is not trivial!

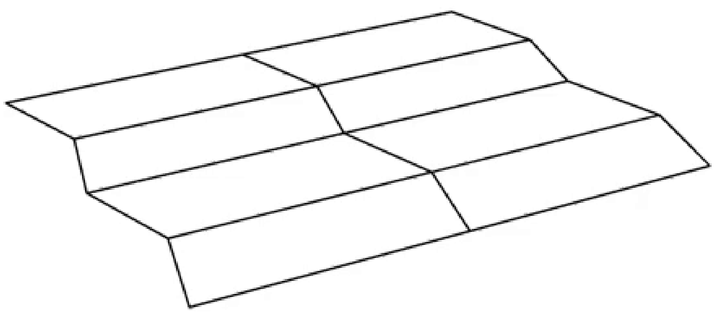


# Rigid-Foldable Tube Basics

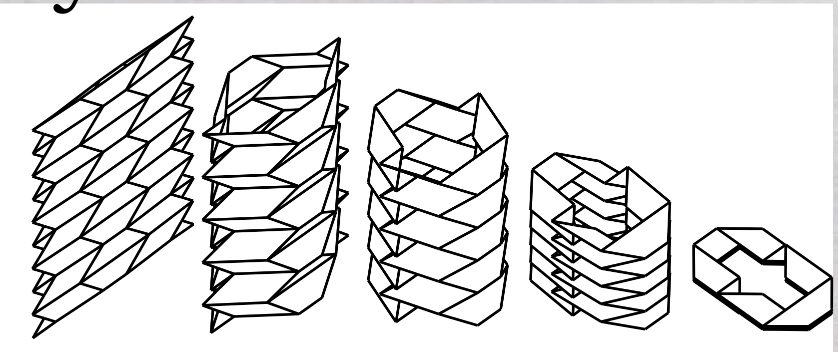
Miura-Ori Reflection

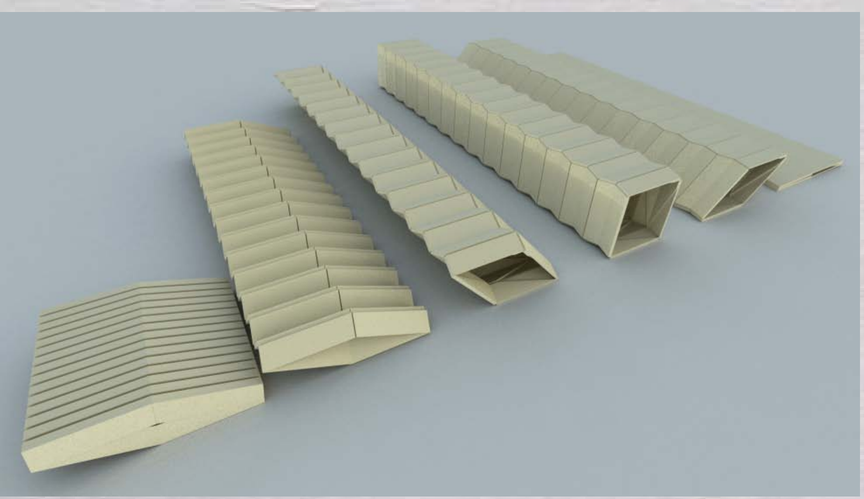
(Partial Structure of
Thoki Yenn's "Flip Flop")

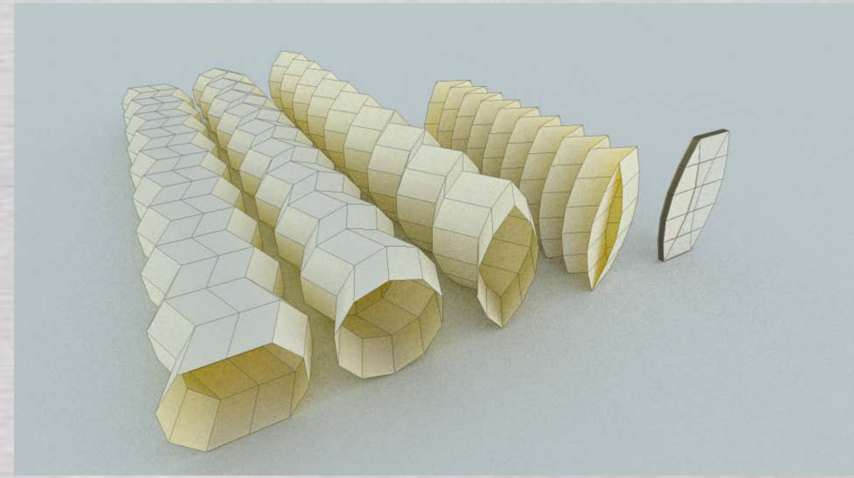


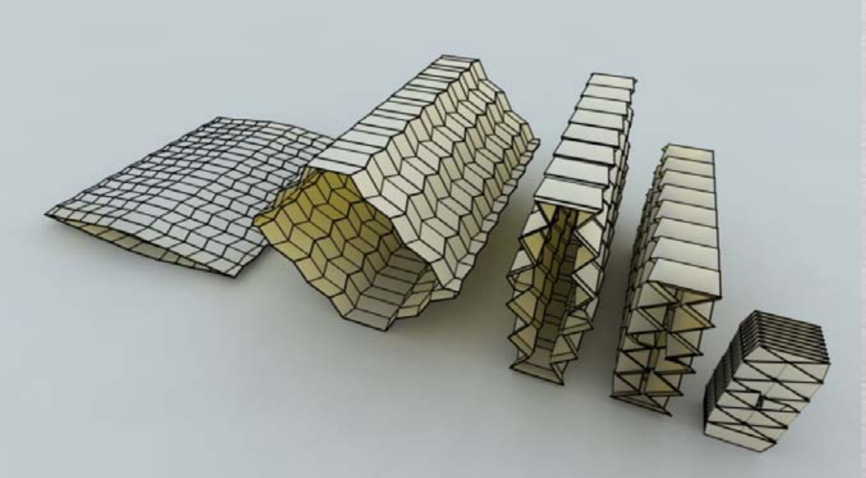



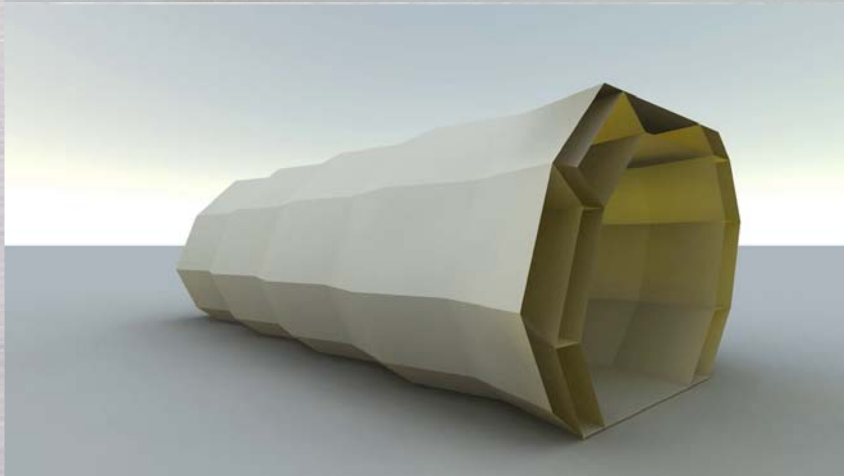
# Symmetric Structure Variations



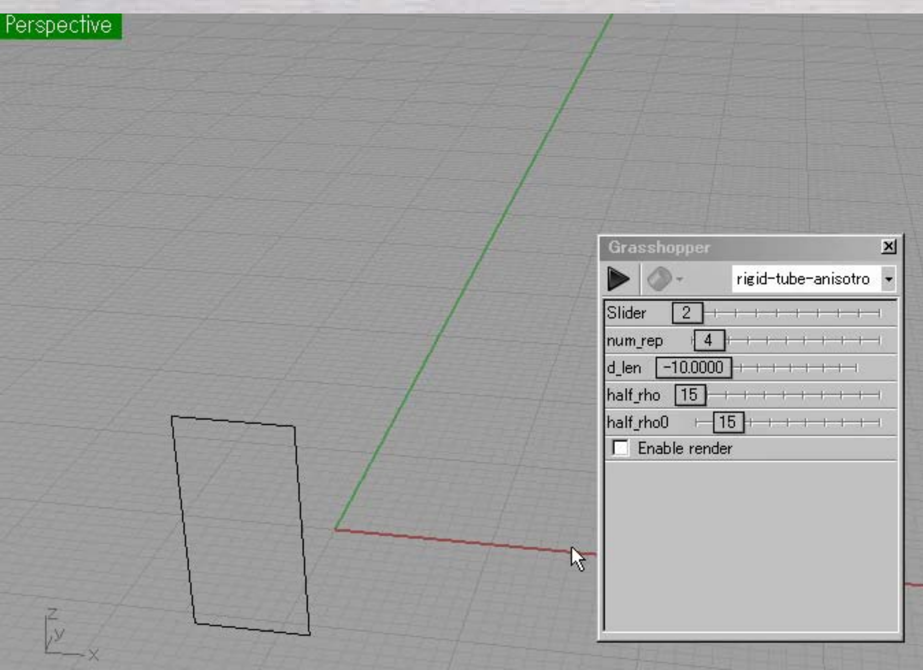


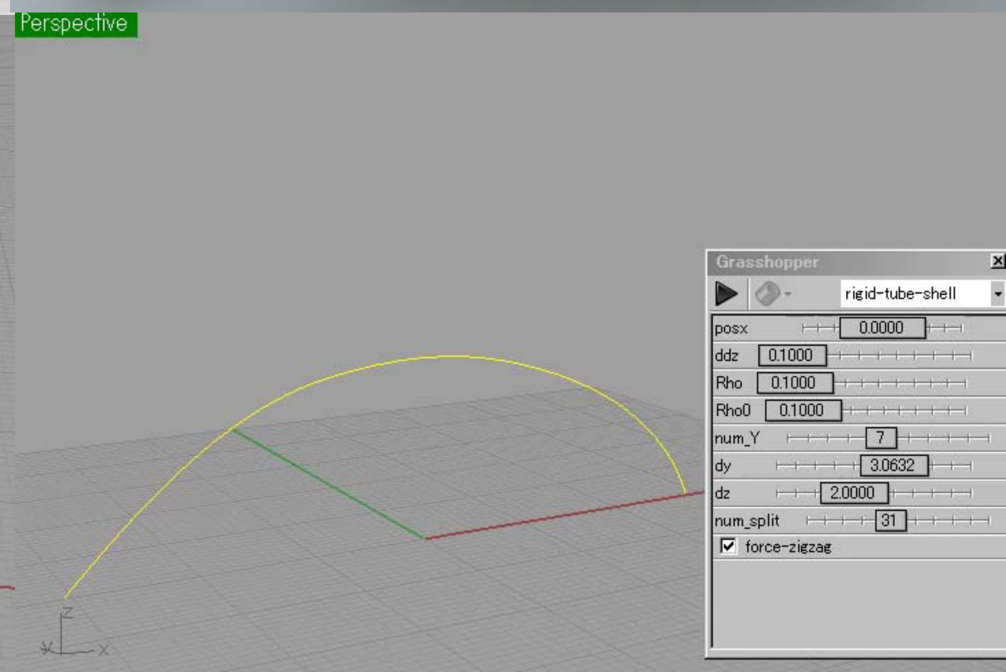




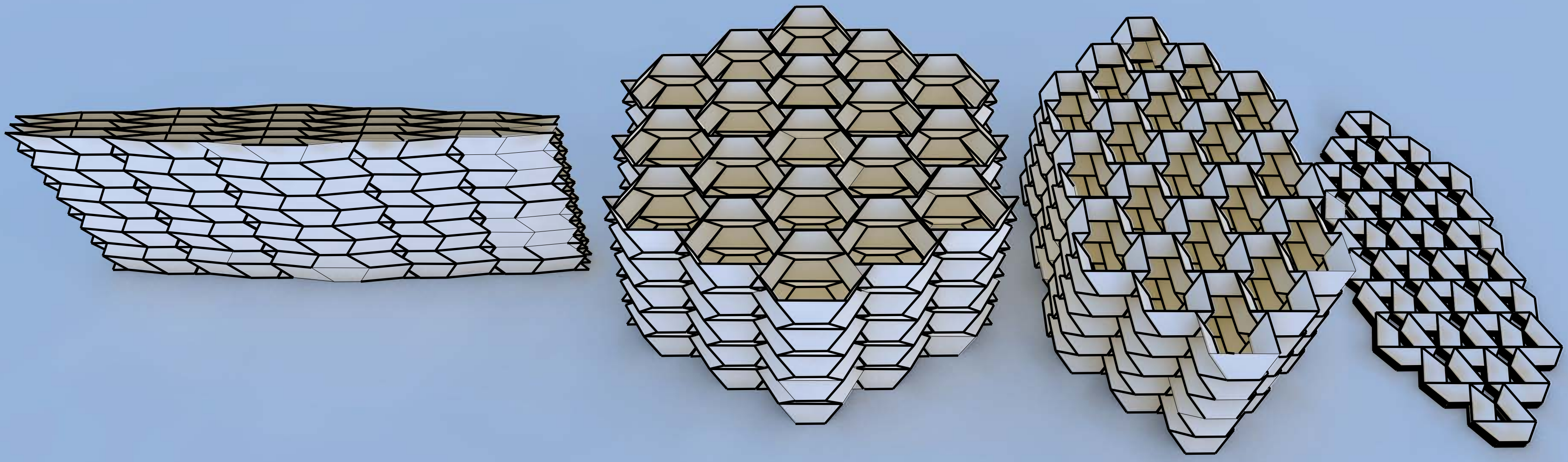


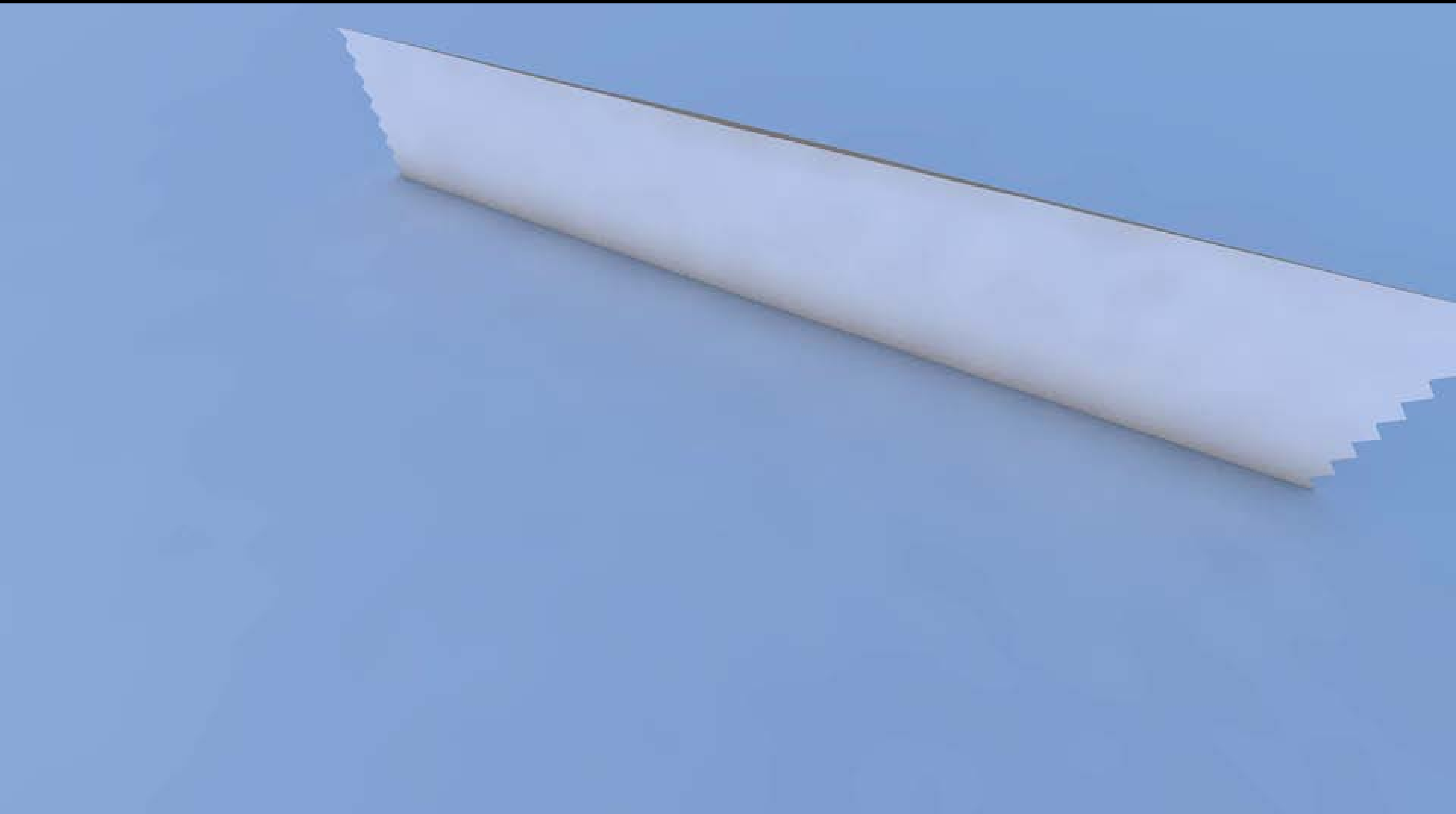
# Parametric design of cylinders and composite structures

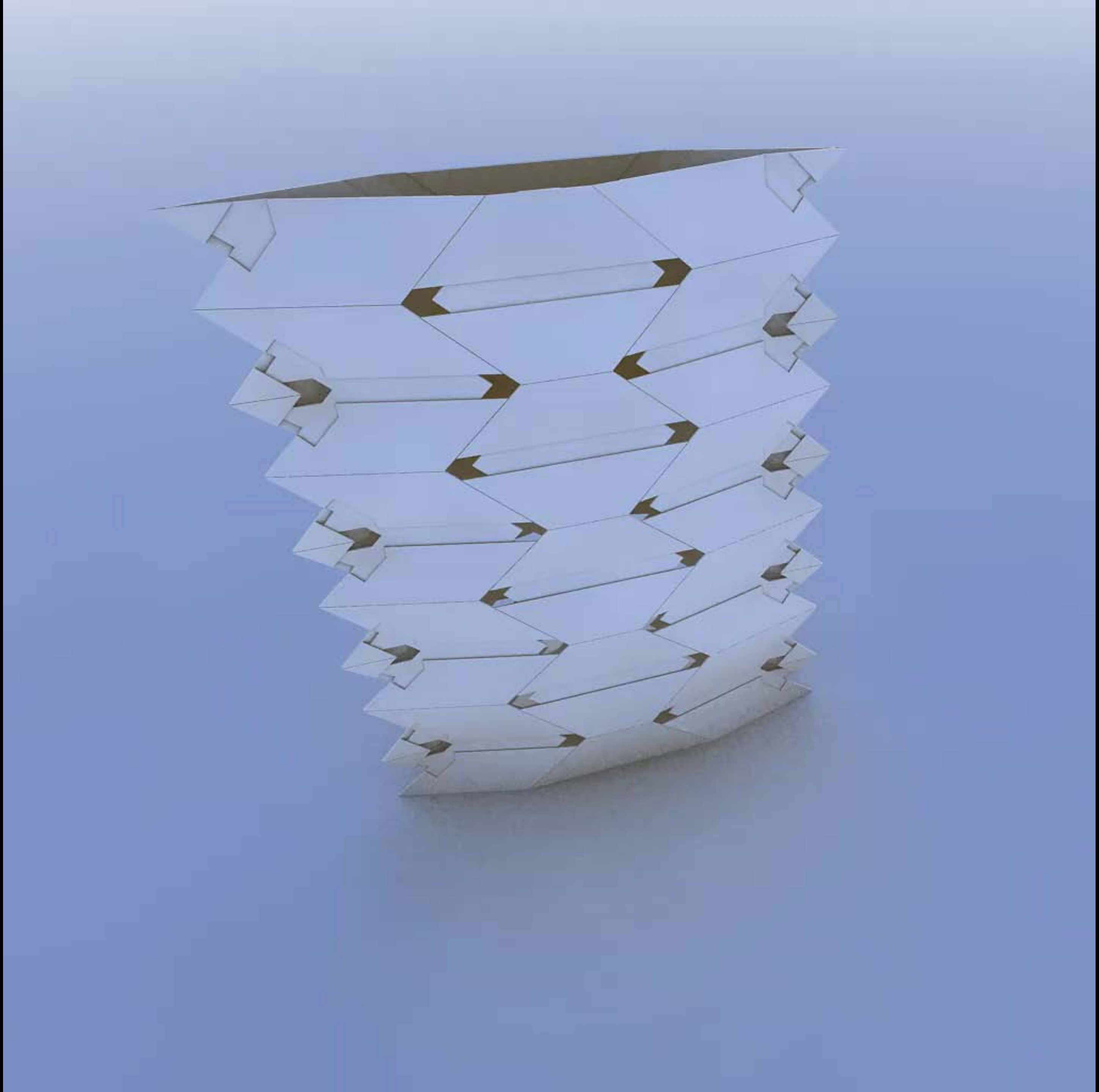


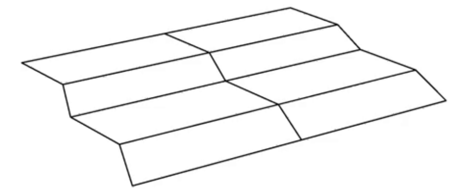
# Cylinder -> Cellular Structure [Miura & Tachi 2010]





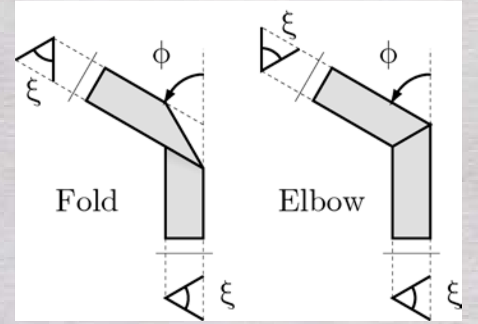


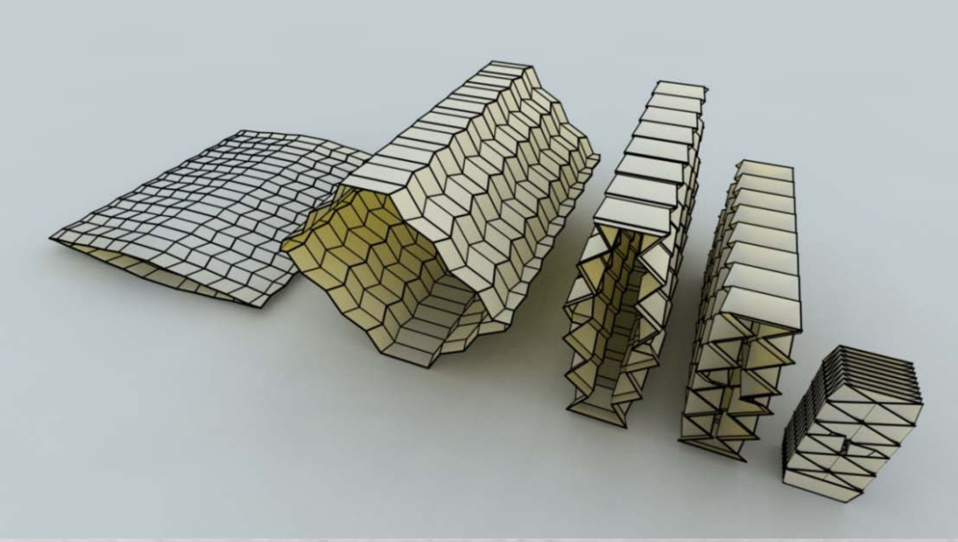
# Isotropic Rigid Foldable Tube Generalization



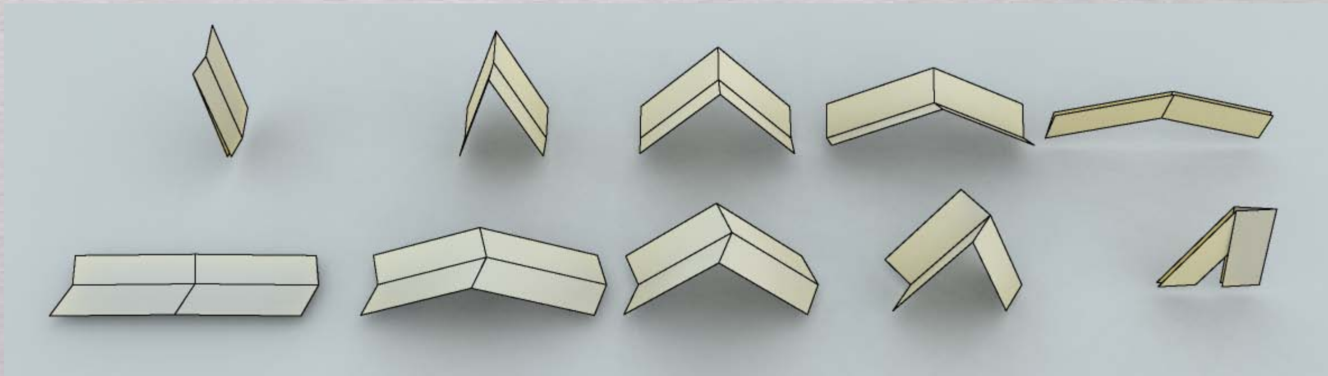
- Rigid Foldable Tube based on symmetry

- Based on
  - "Fold"
  - "Elbow"



= special case of BDFFPQ Mesh



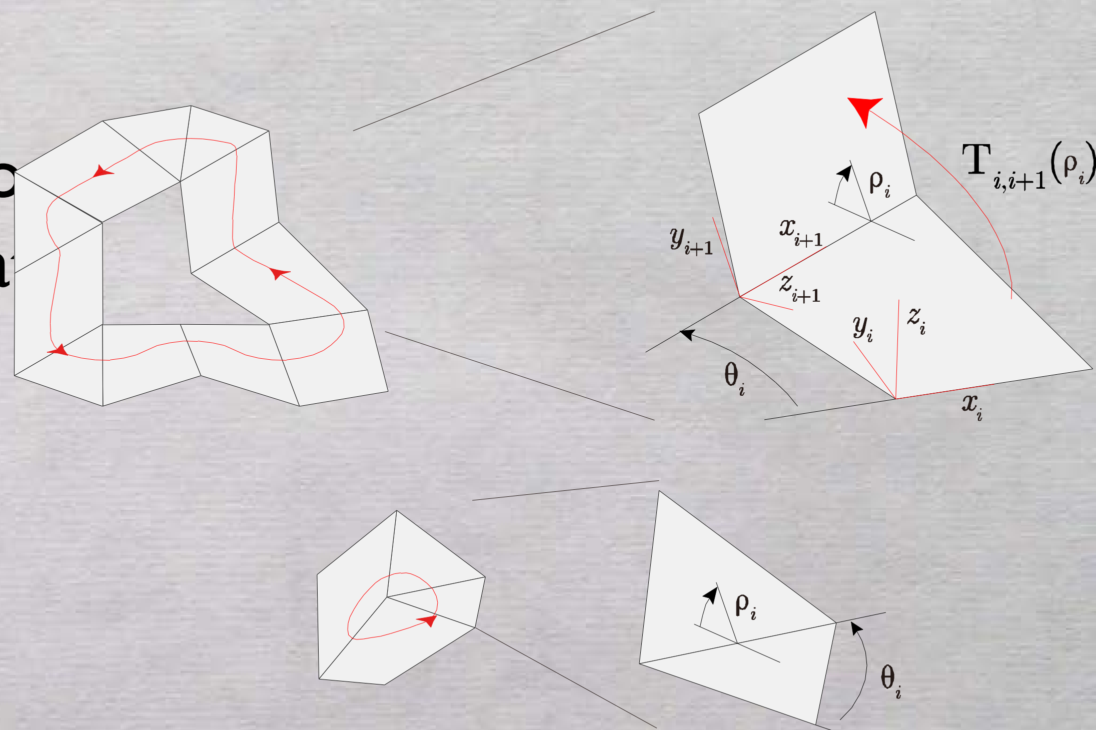
# Generalized Rigid Folding Constraints

- For any closed loop in Mesh

$$T_{0,1} \cdots T_{k-2,k-1} T_{k-1,0} = \mathbf{I}$$

where $T_{i,j}$ is a 4x4 transformation matrix to translate facets coordina i to j

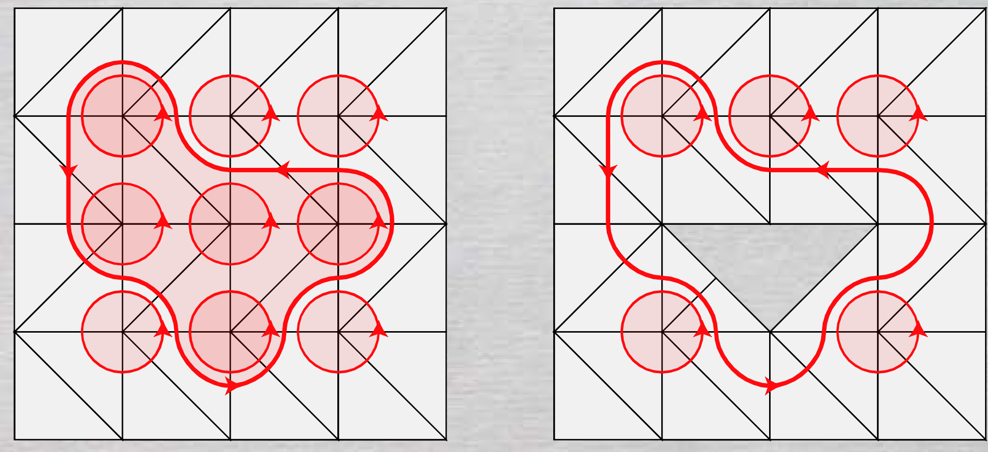
- When it is around a vertex: T is a rotation matrix.

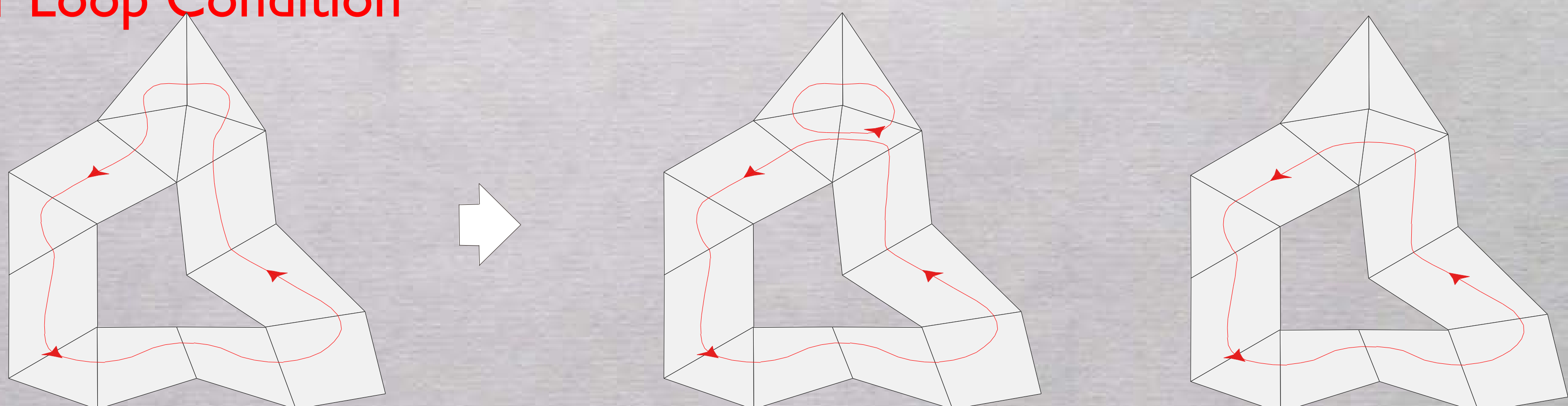


# Generalized Rigid Folding Constraints

- If the loop surrounds no hole:
  - constraints around each vertex
- If there is a hole,
  - constraints around each vertex

  + 1 Loop Condition

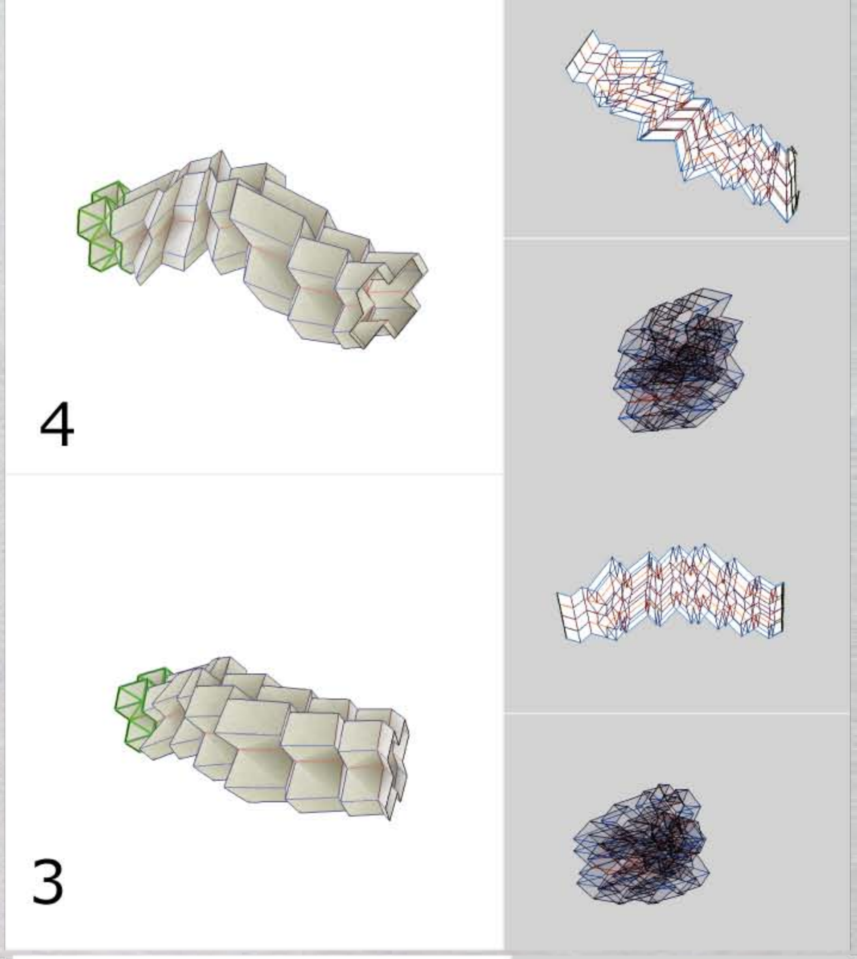


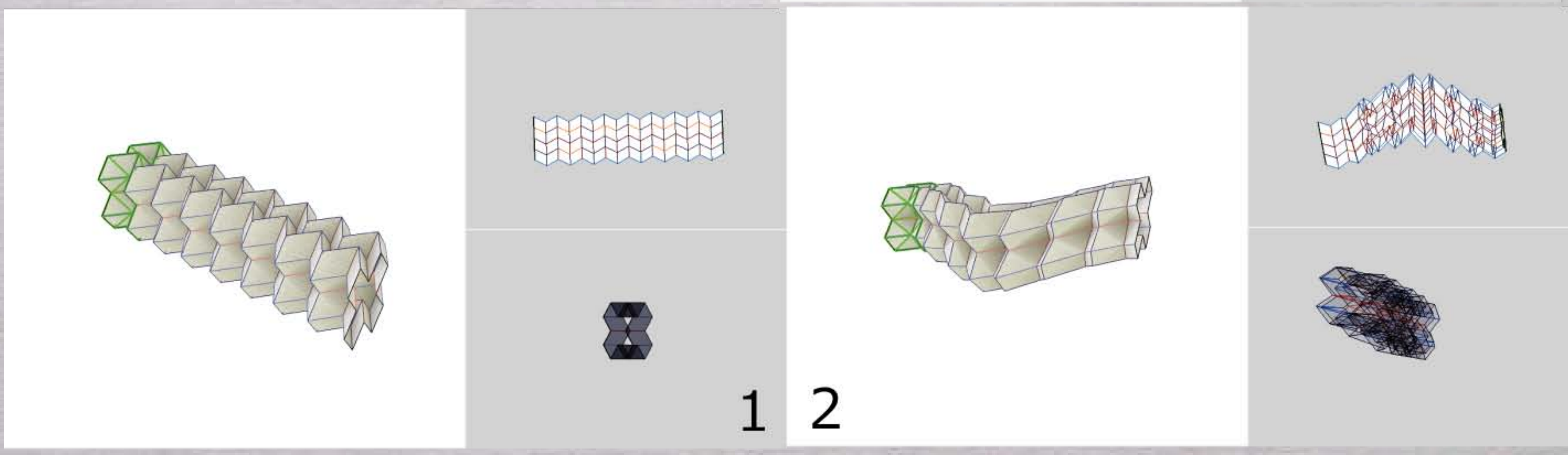


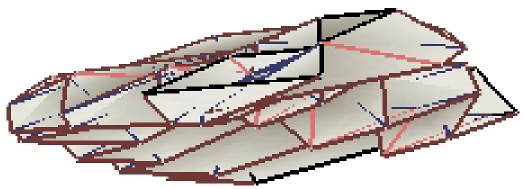
# Loop Condition
# : Sufficient Condition

loop condition for finite rigid foldability

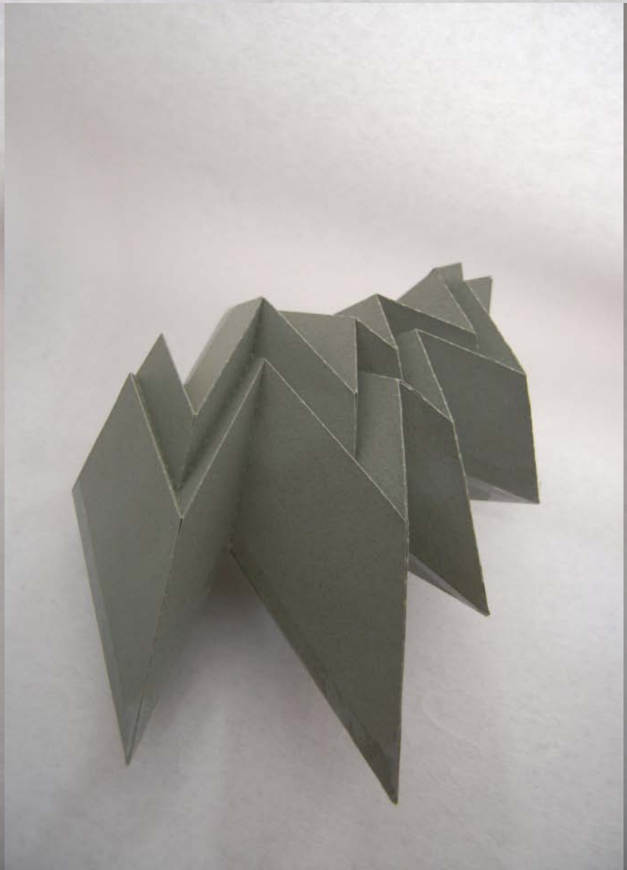
→ Sufficient Condition
: start from symmetric cylinder and fix 1 loop

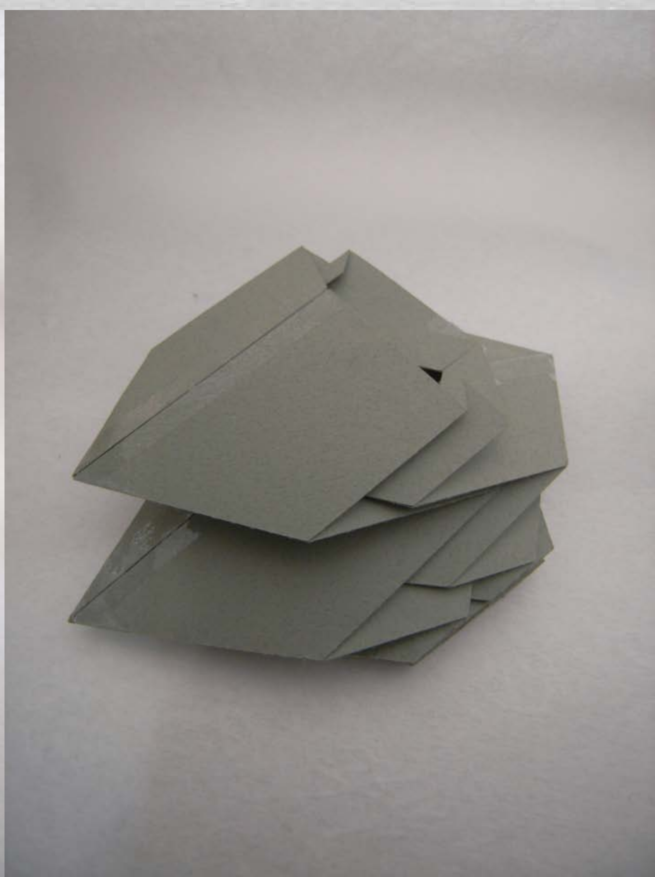


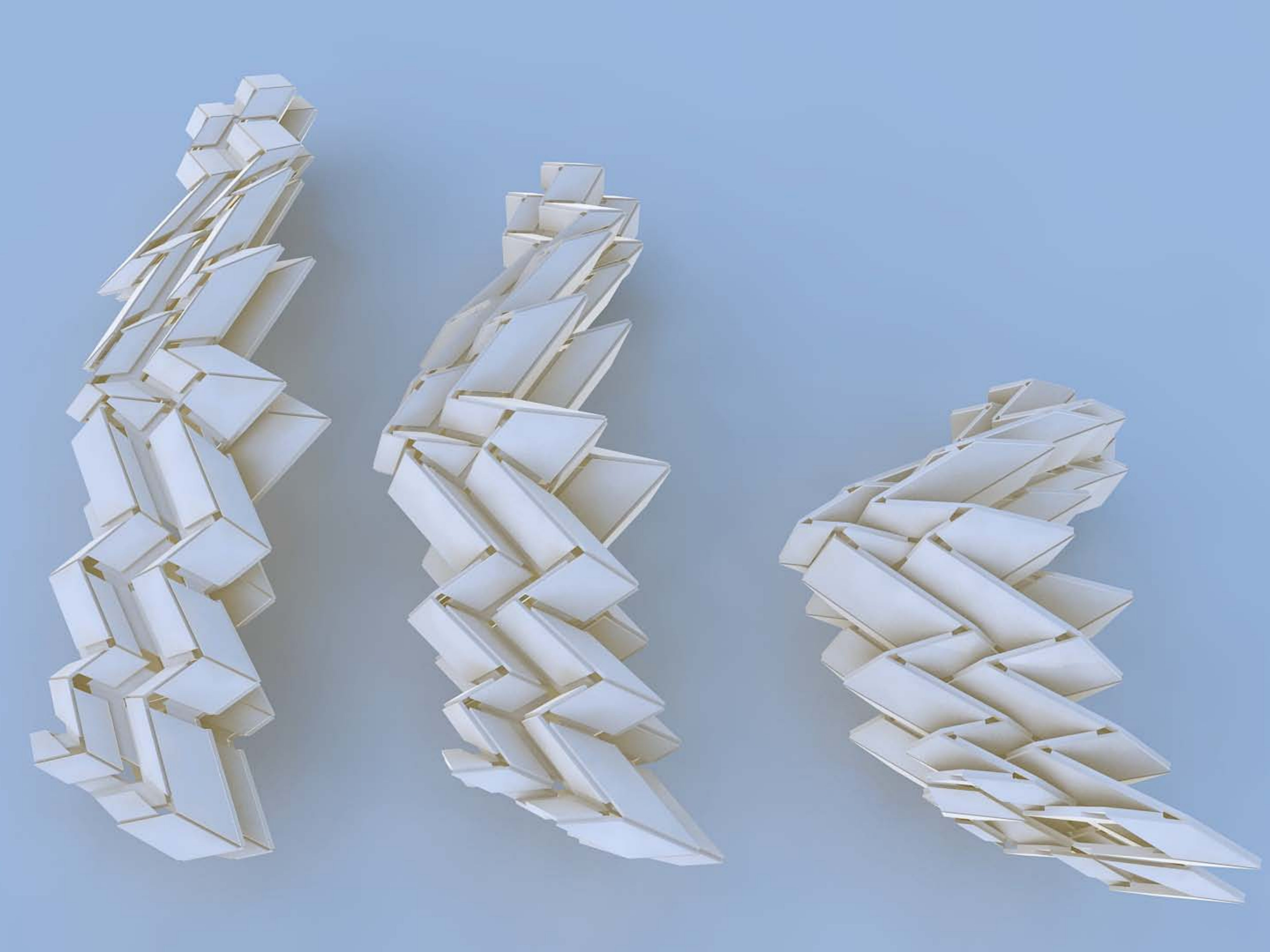



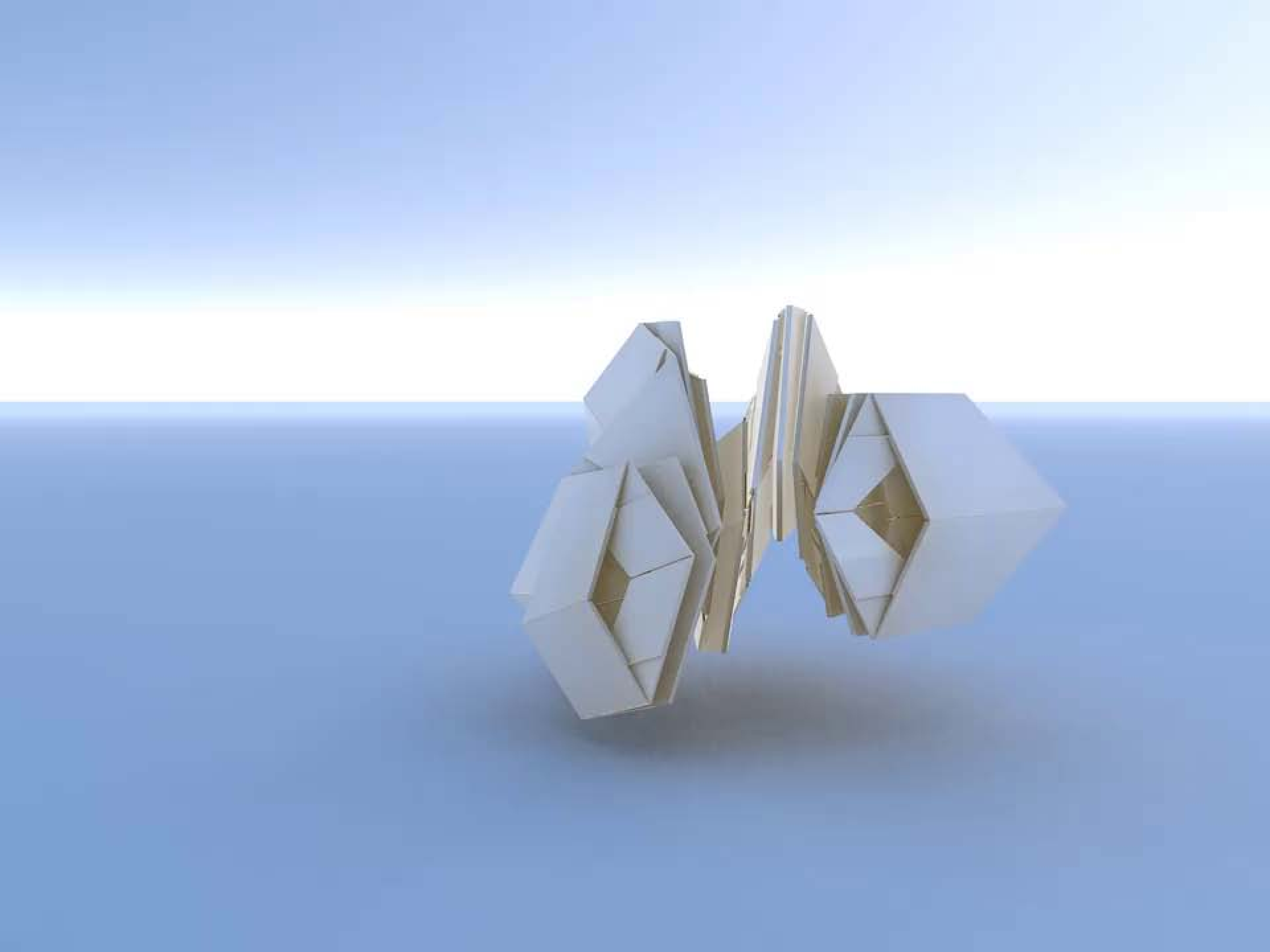


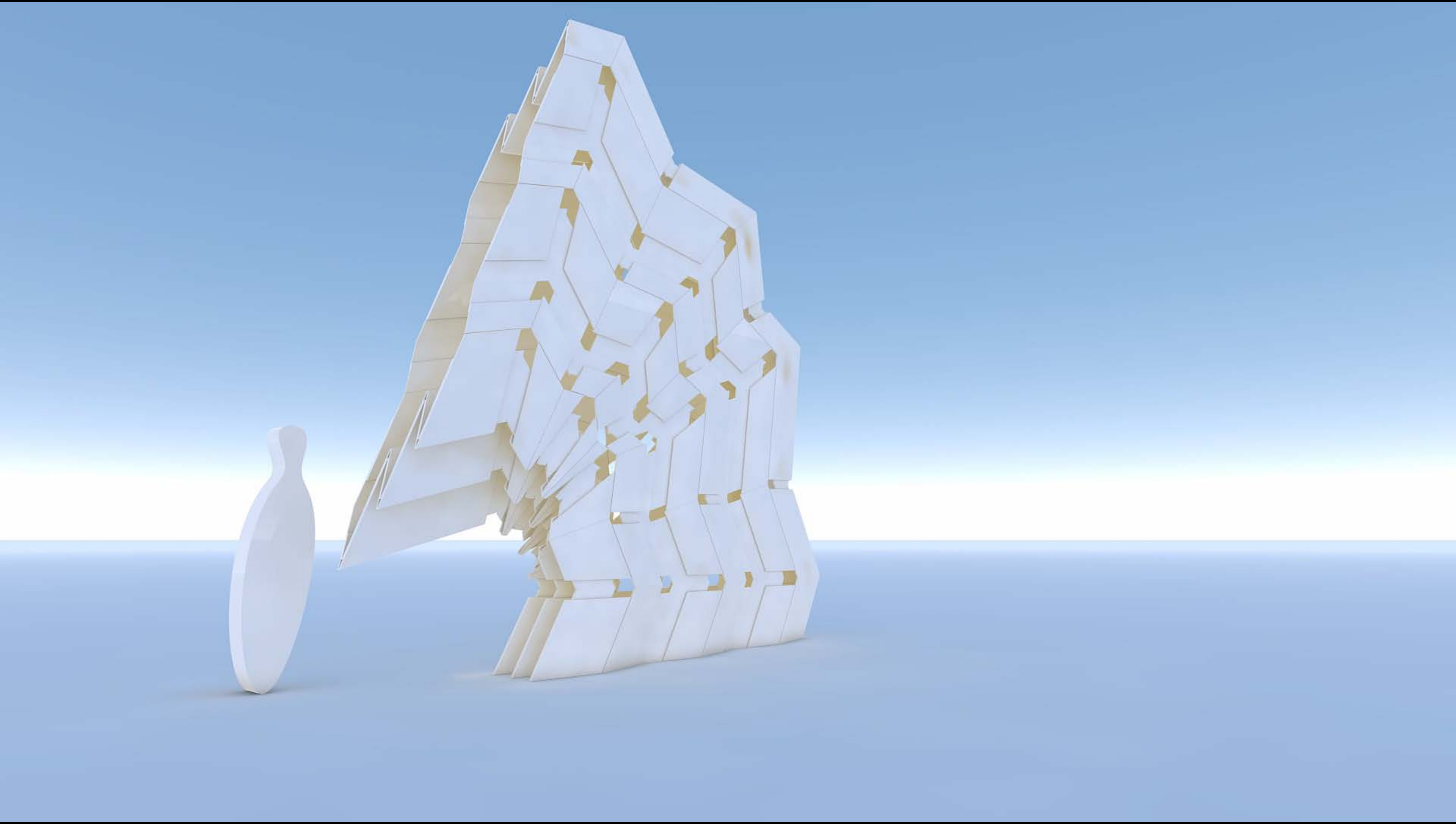
# Manufactured From Two Sheets of Paper

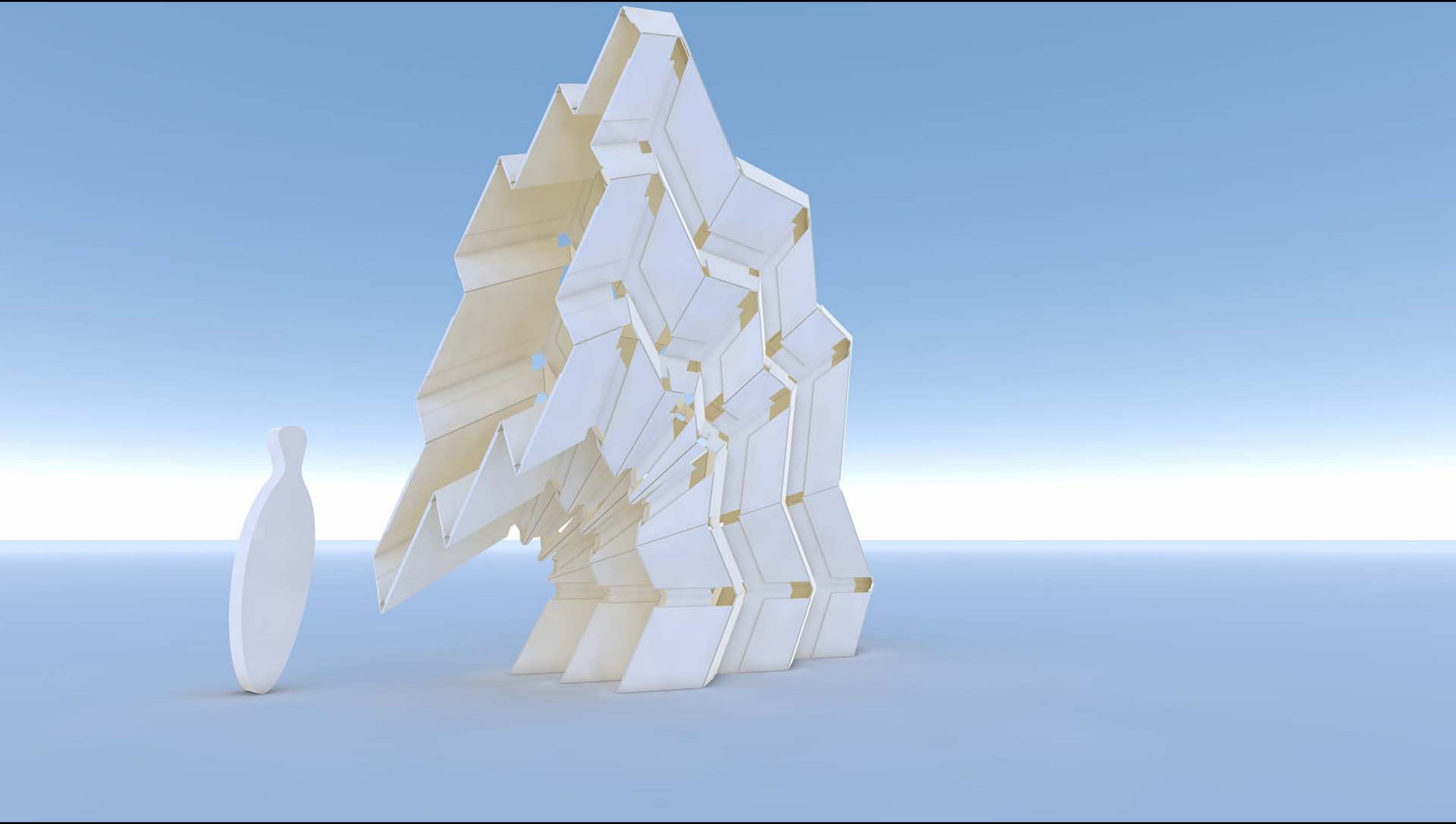


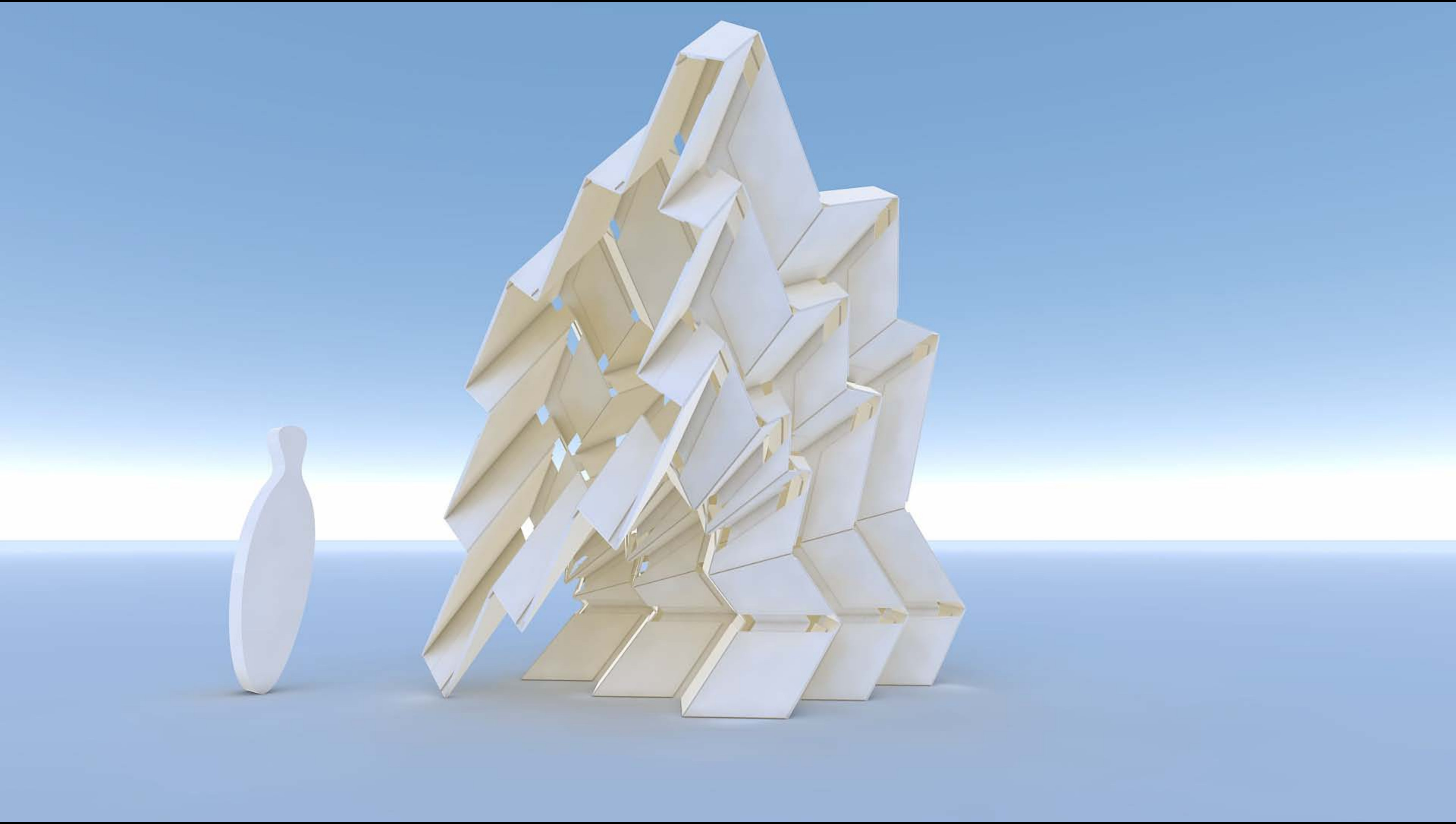


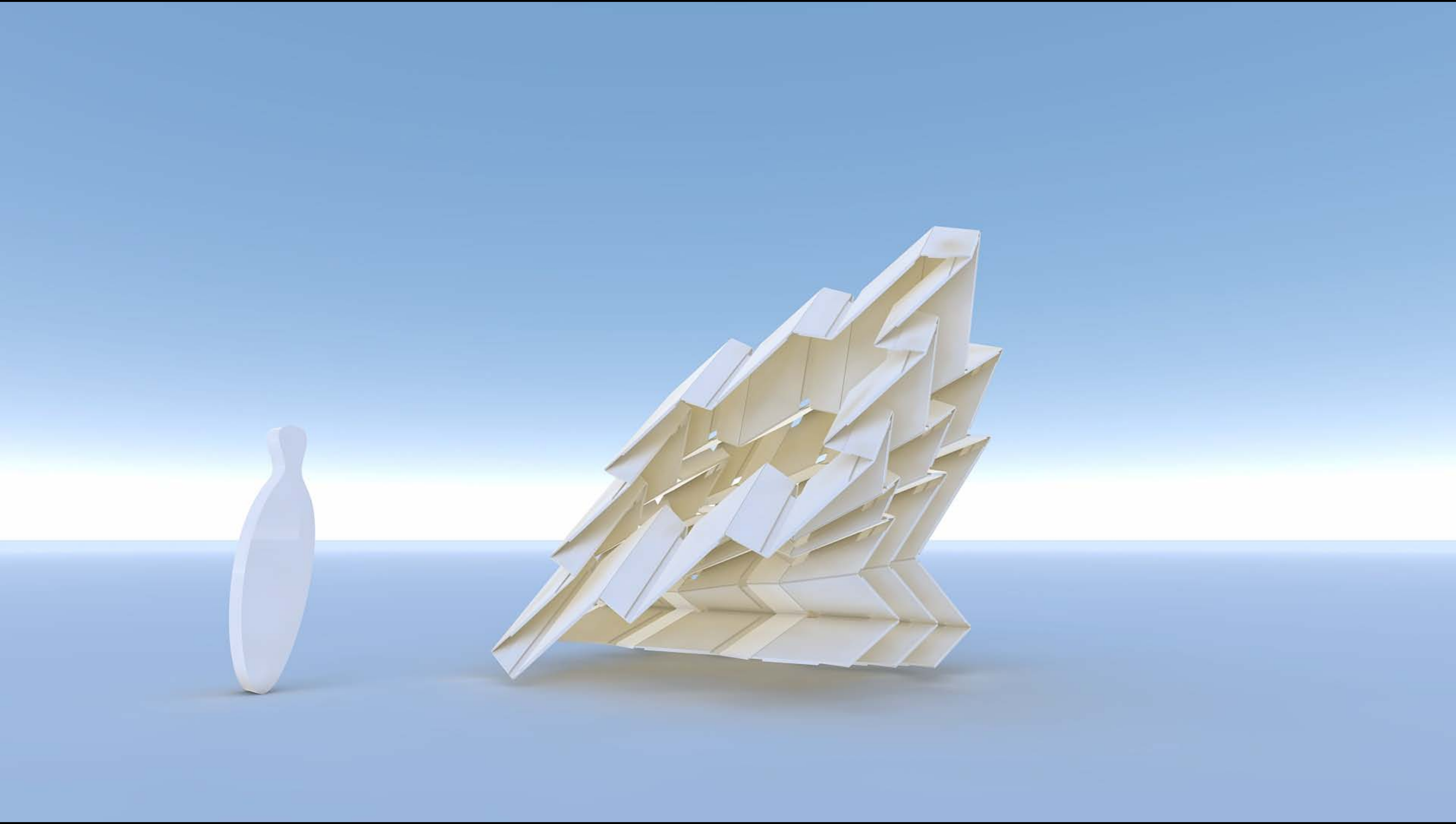


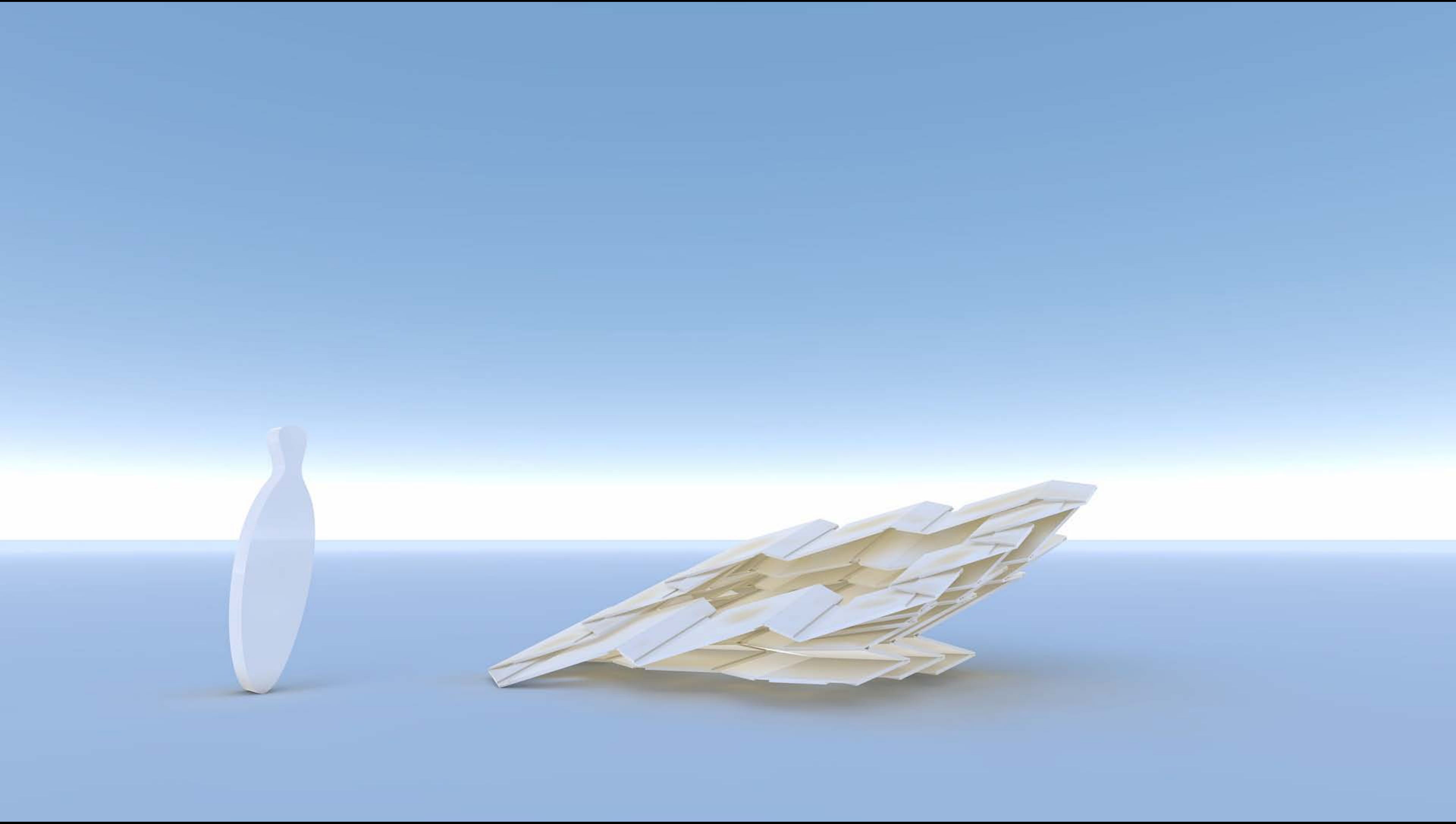






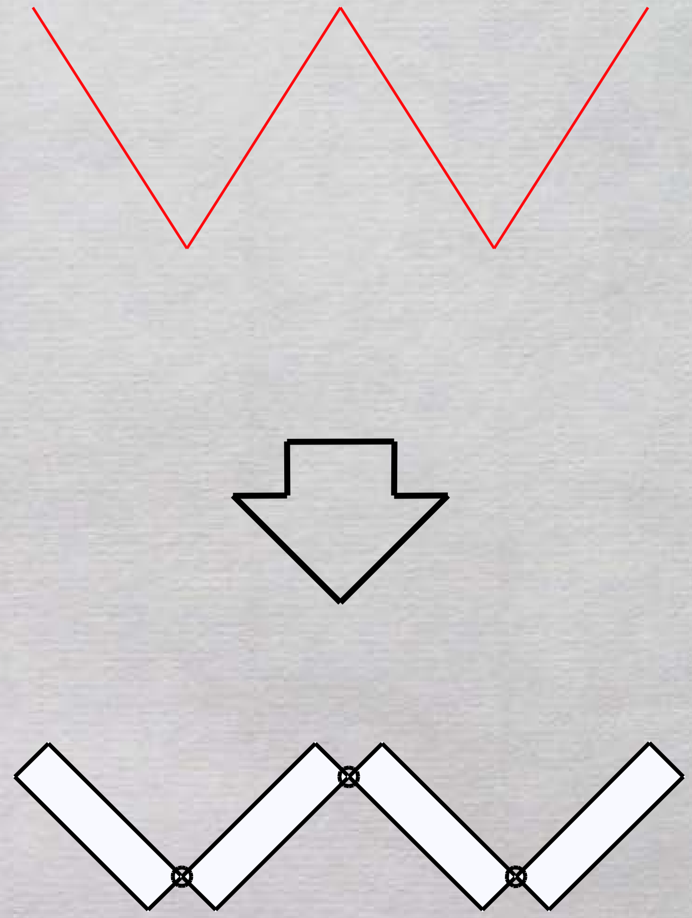




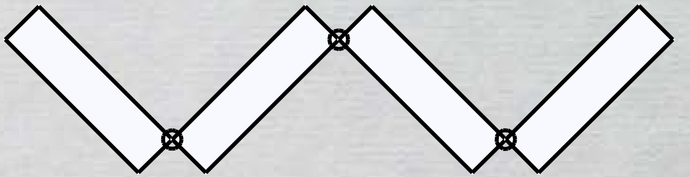


# Thickening

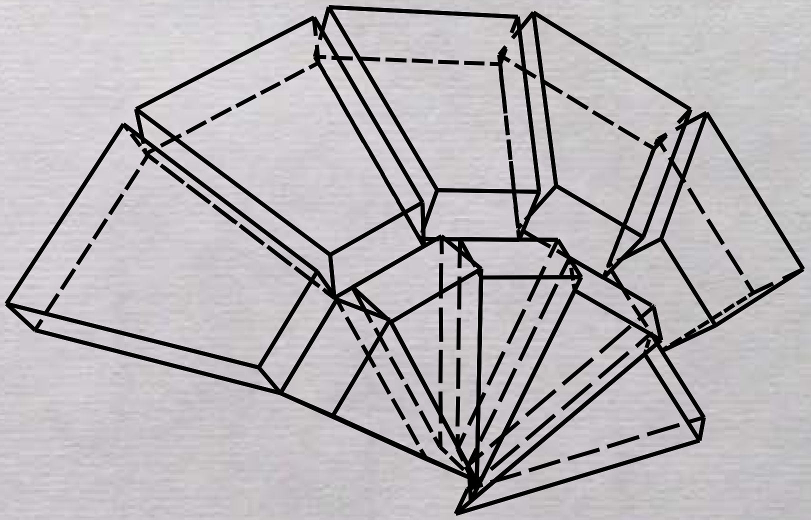
- Rigid origami is ideal surface (no thickness)
- Reality:
  - There is thickness
  - To make "rigid" panels, thickness must be solved geometrically
- Modified Model:
  - Thick plates
  - Rotating hinges at the edges



# Hinge Shift Approach

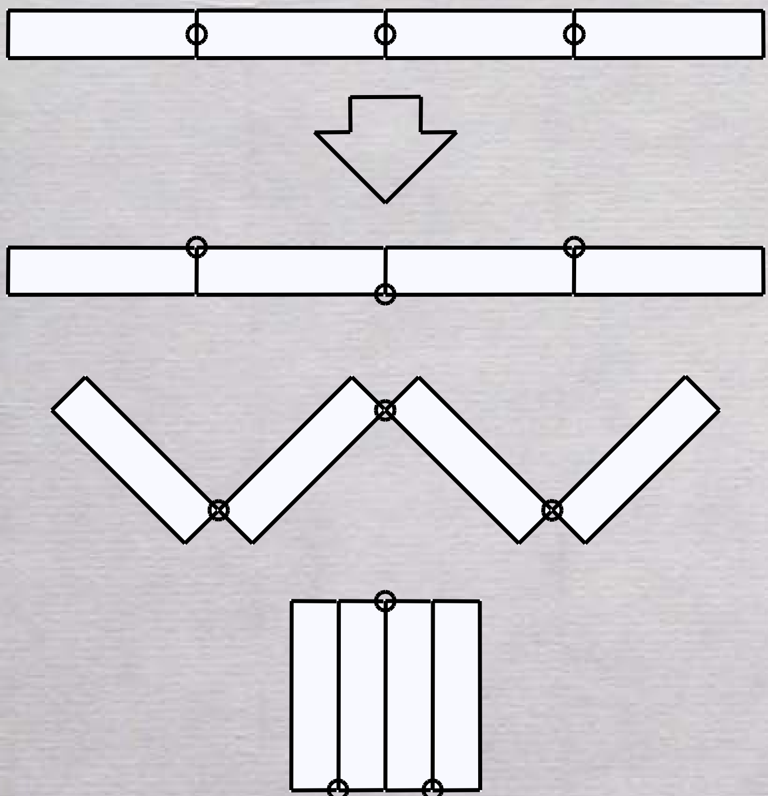


- Main Problem
  - non-concurrent edges →6 constraints (overconstrained)
- Symmetric Vertex:
  - [Hoberman 88]
  - use two levels of thickness
  - works only if the vertex is symmetric (a = b, c=d=π-a)
- Slidable Hinges
  - [Trautz and Kunstler 09]
  - Add extra freedom by allowing „slide“
  - Problem: global accumulation of slide (not locally designable)



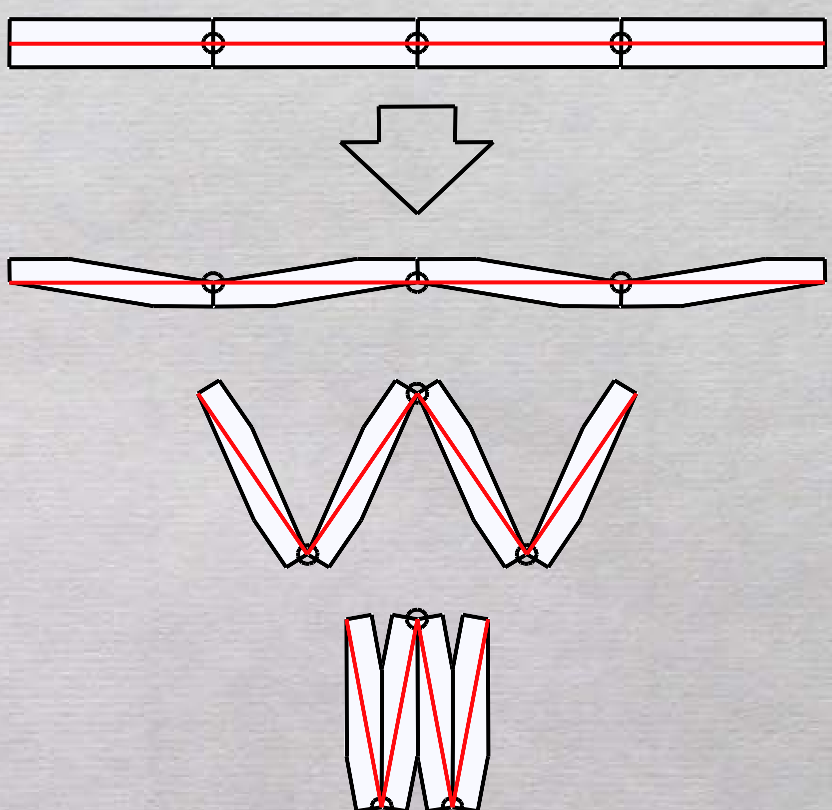
# Our Approach

**Hinge Shift**



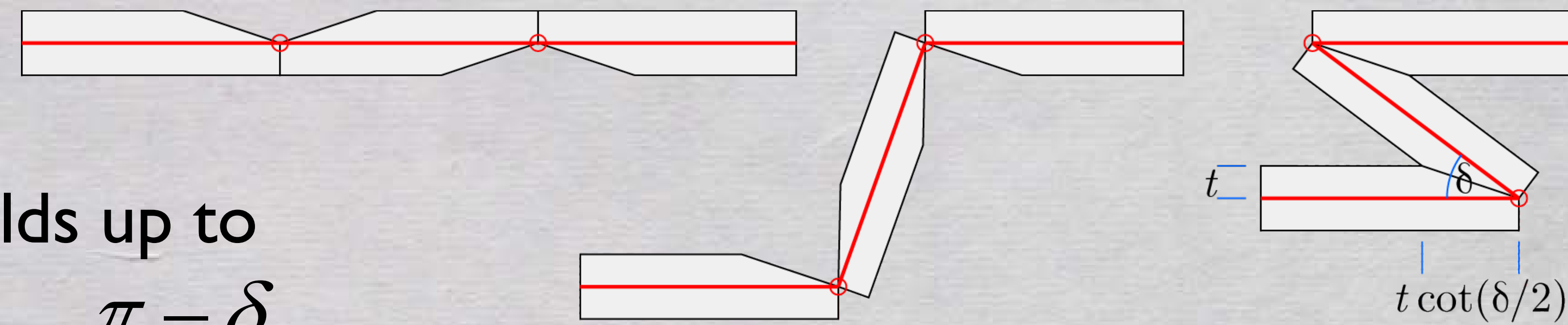
Non-concurrent edges

**Volume Trim**



Concurrent edges

# Trimming Volume

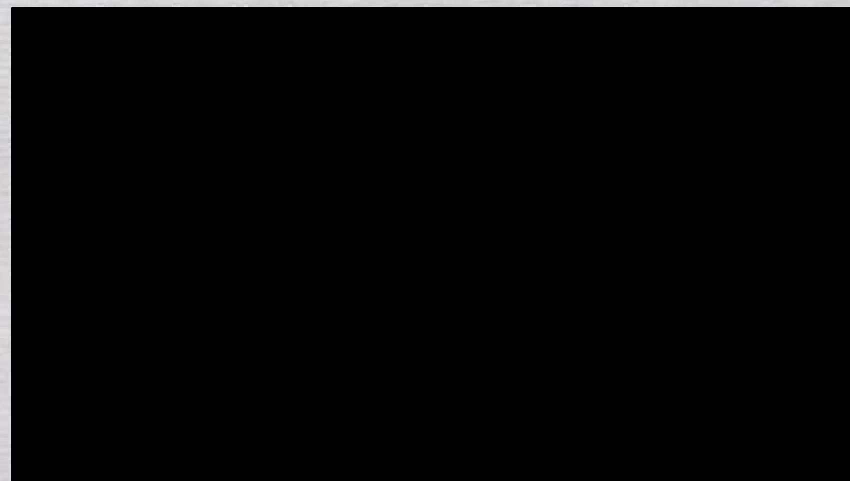



- folds up to

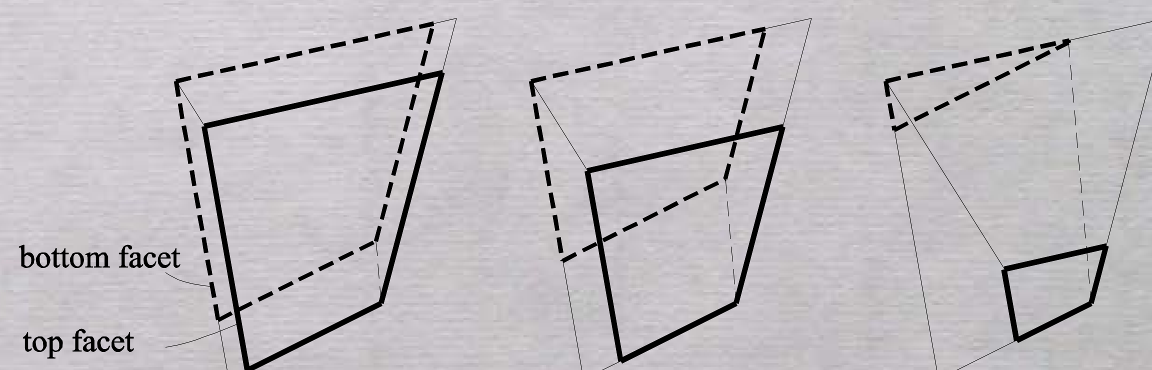
$$\pi - \delta$$

- offsetting edges by

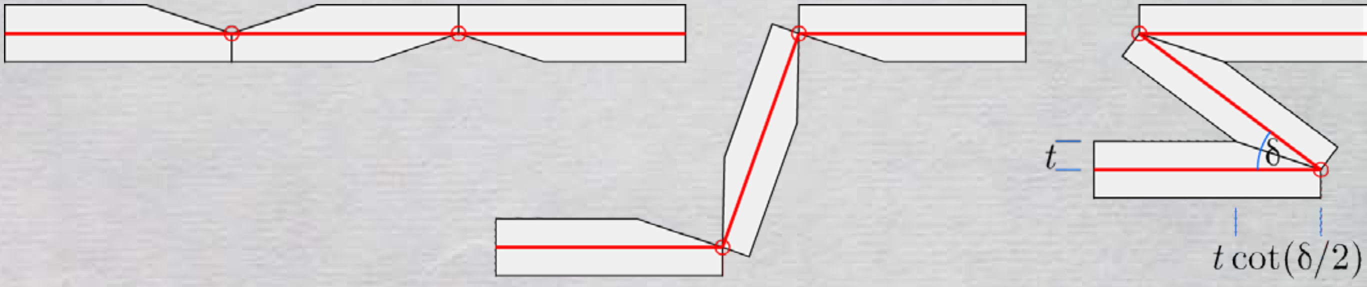
$$t\cot\left(\frac{\delta}{2}\right)$$

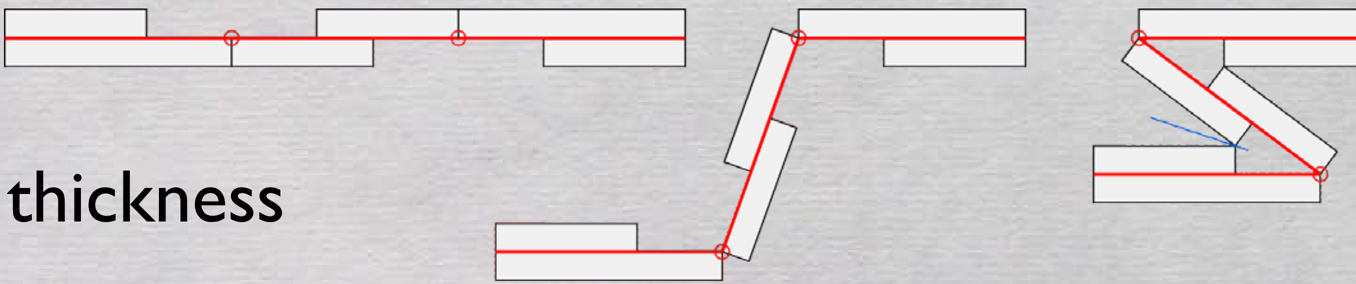


→Different speed for each edge: Weighed Straight Skeleton

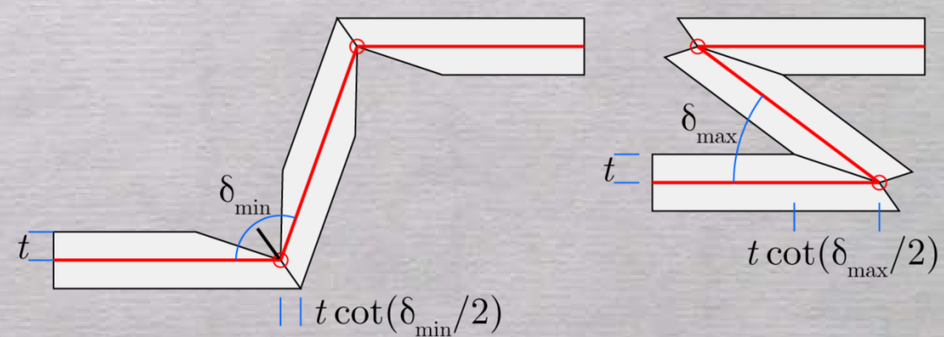


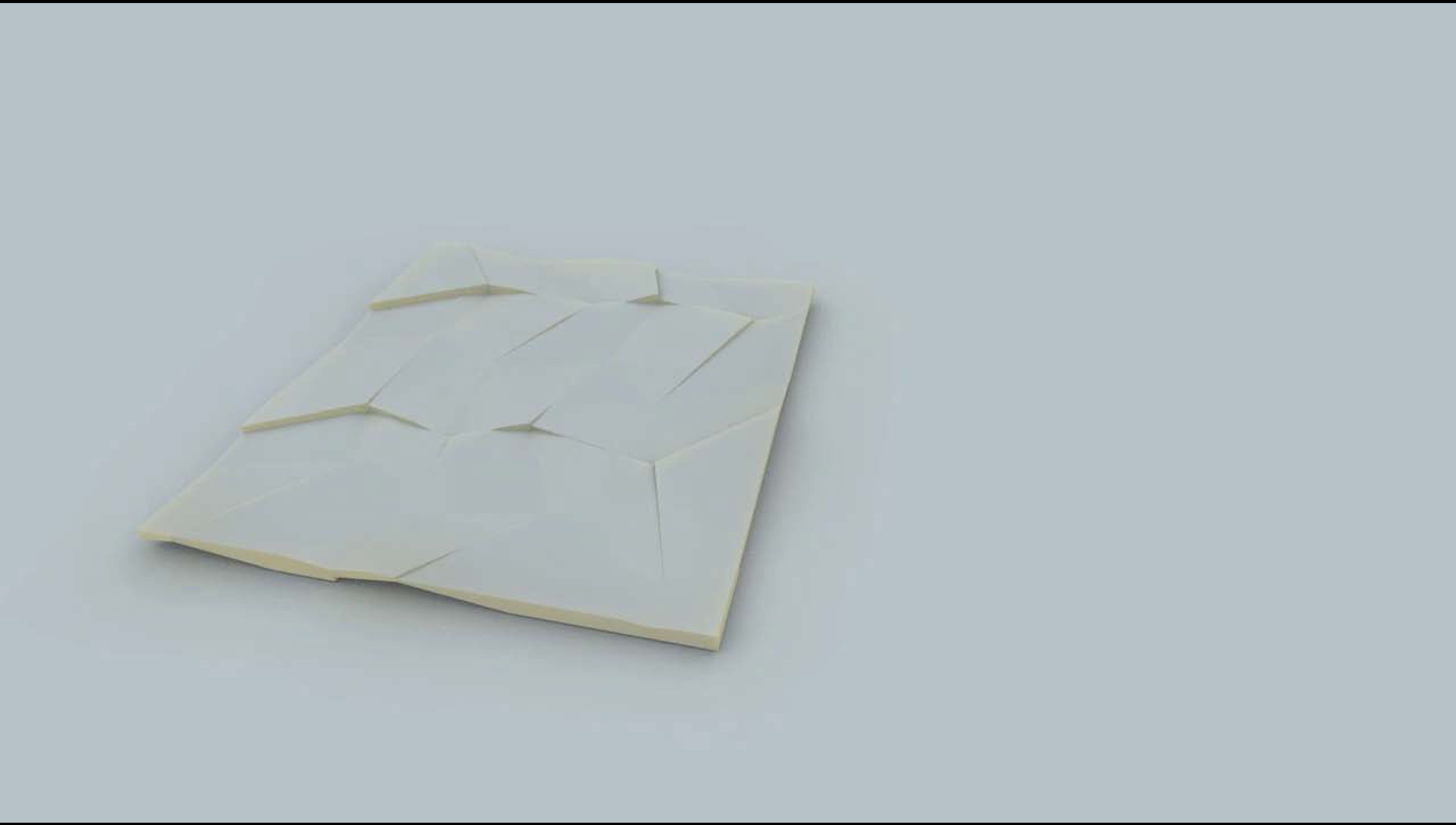
# Variations



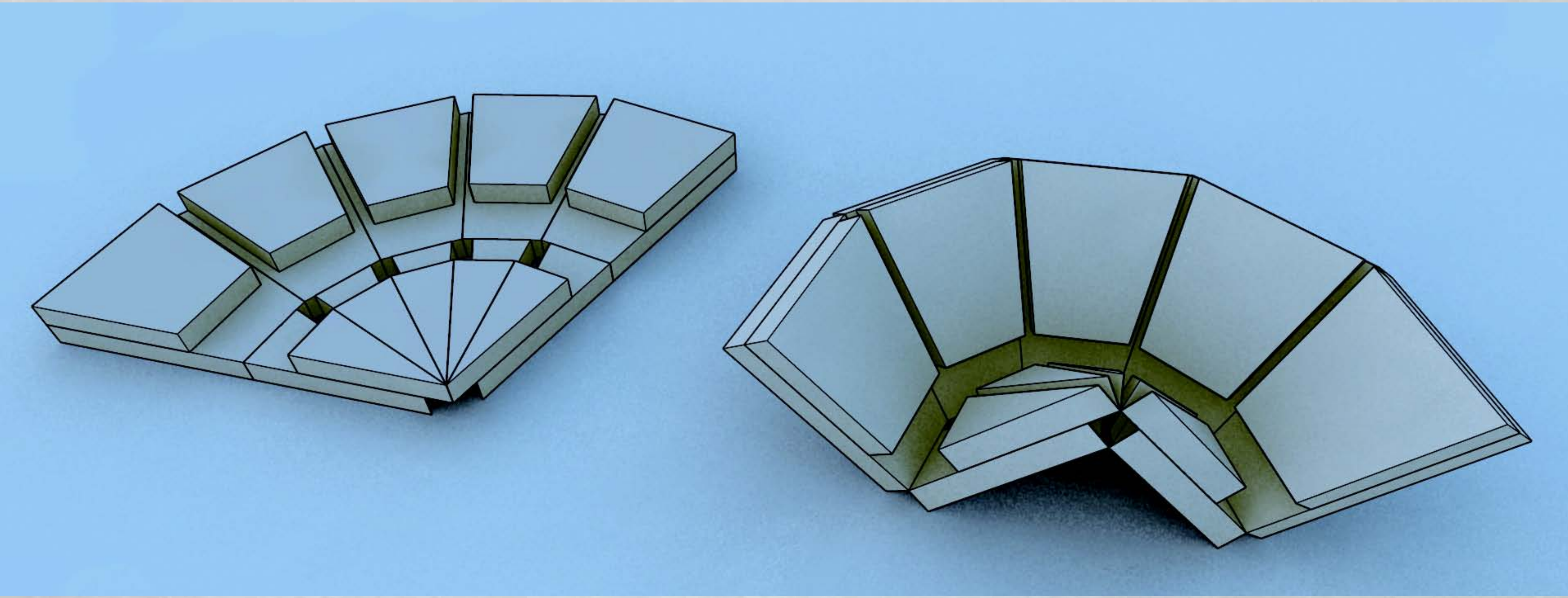



- Use constant thickness panels
  - if both layers overlap sufficiently
- use angle limitation
  - useful for defining the "deployed 3D state"

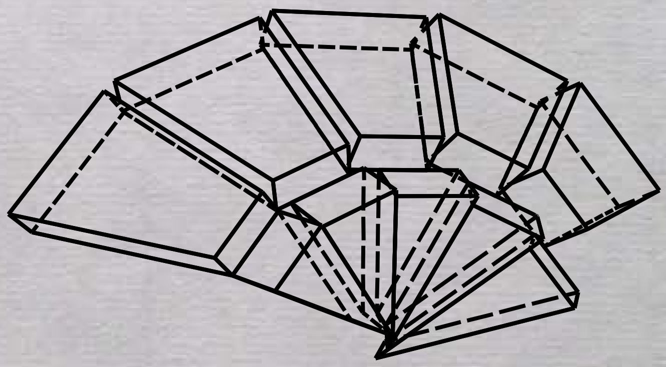


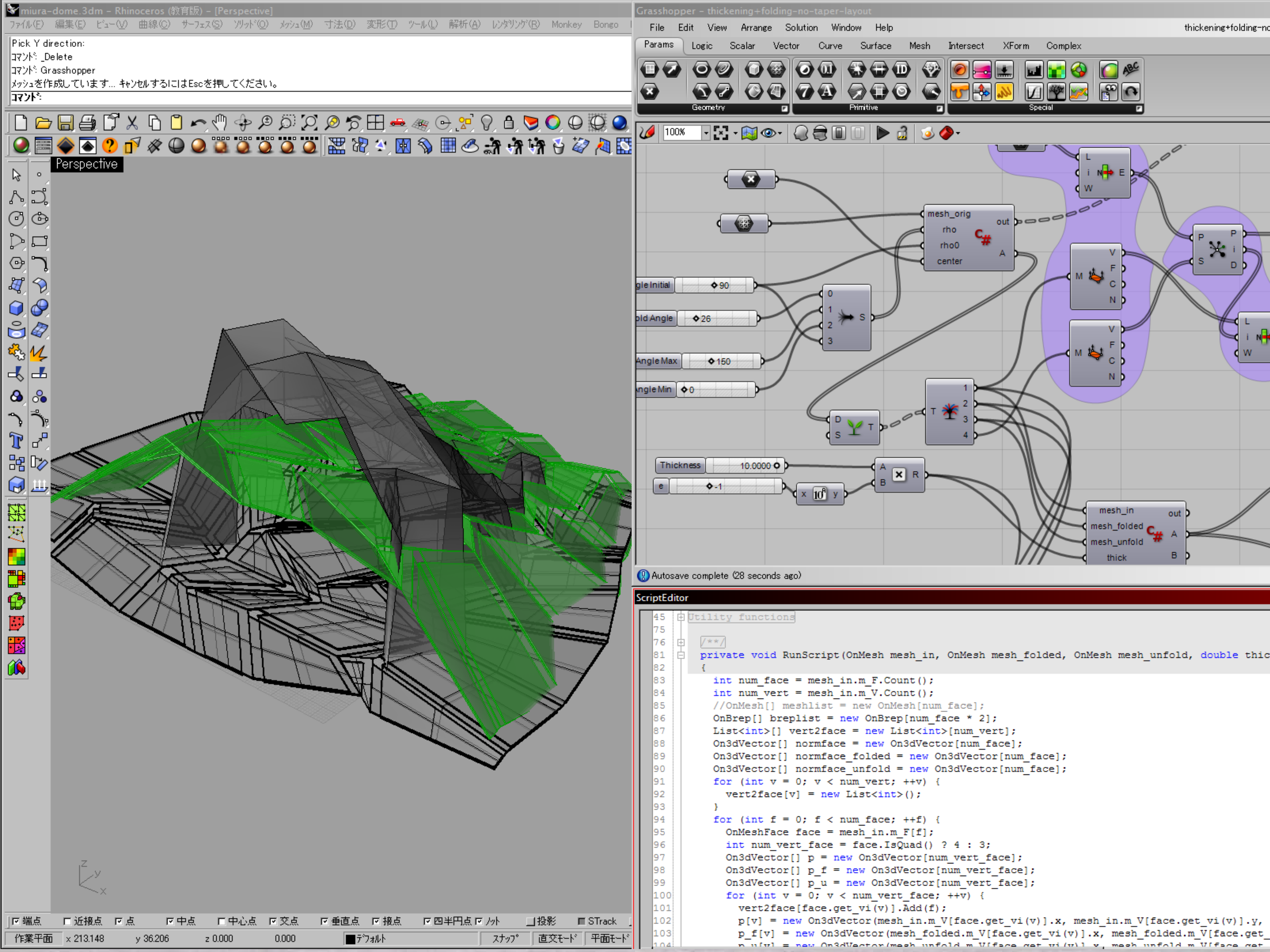


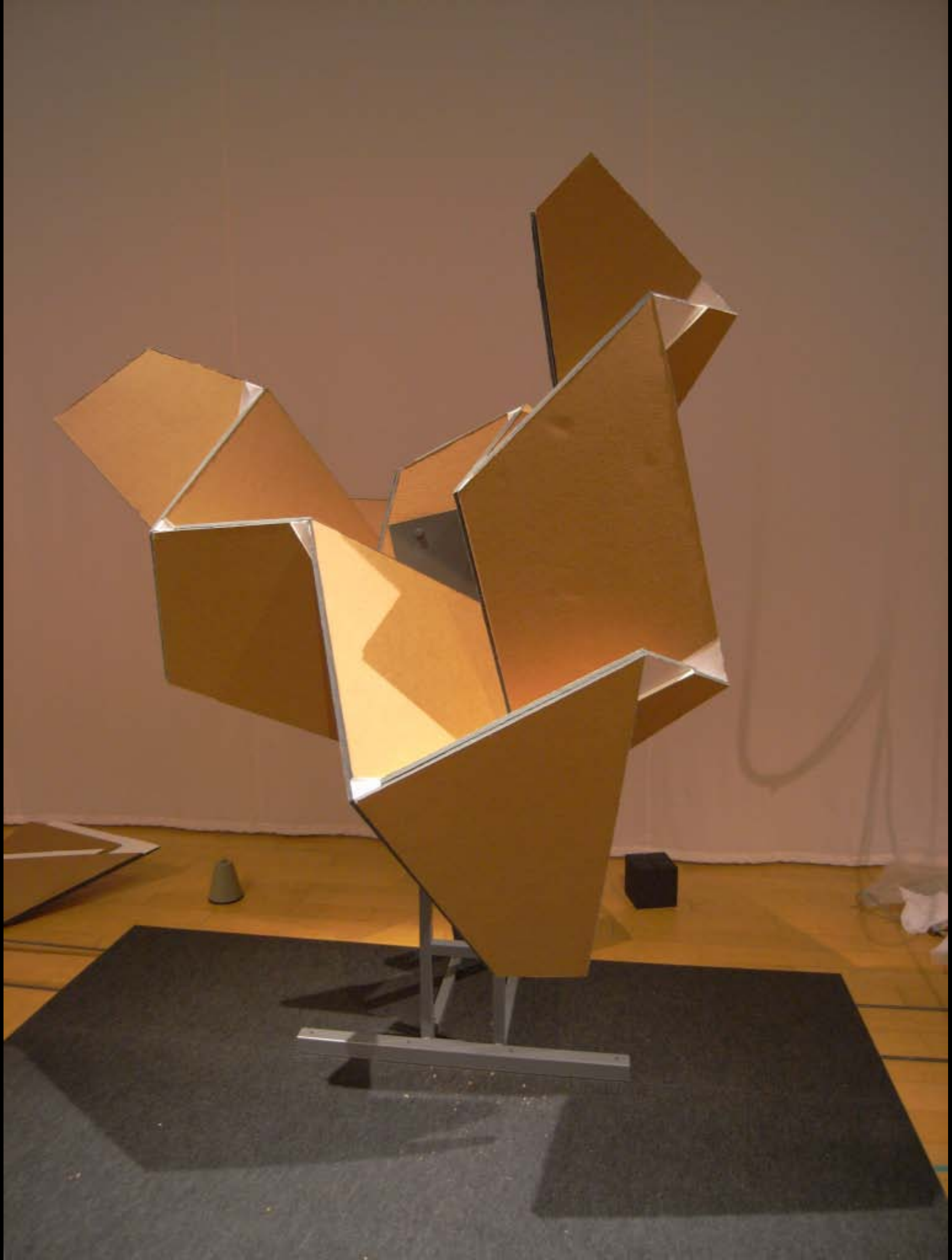
# Example

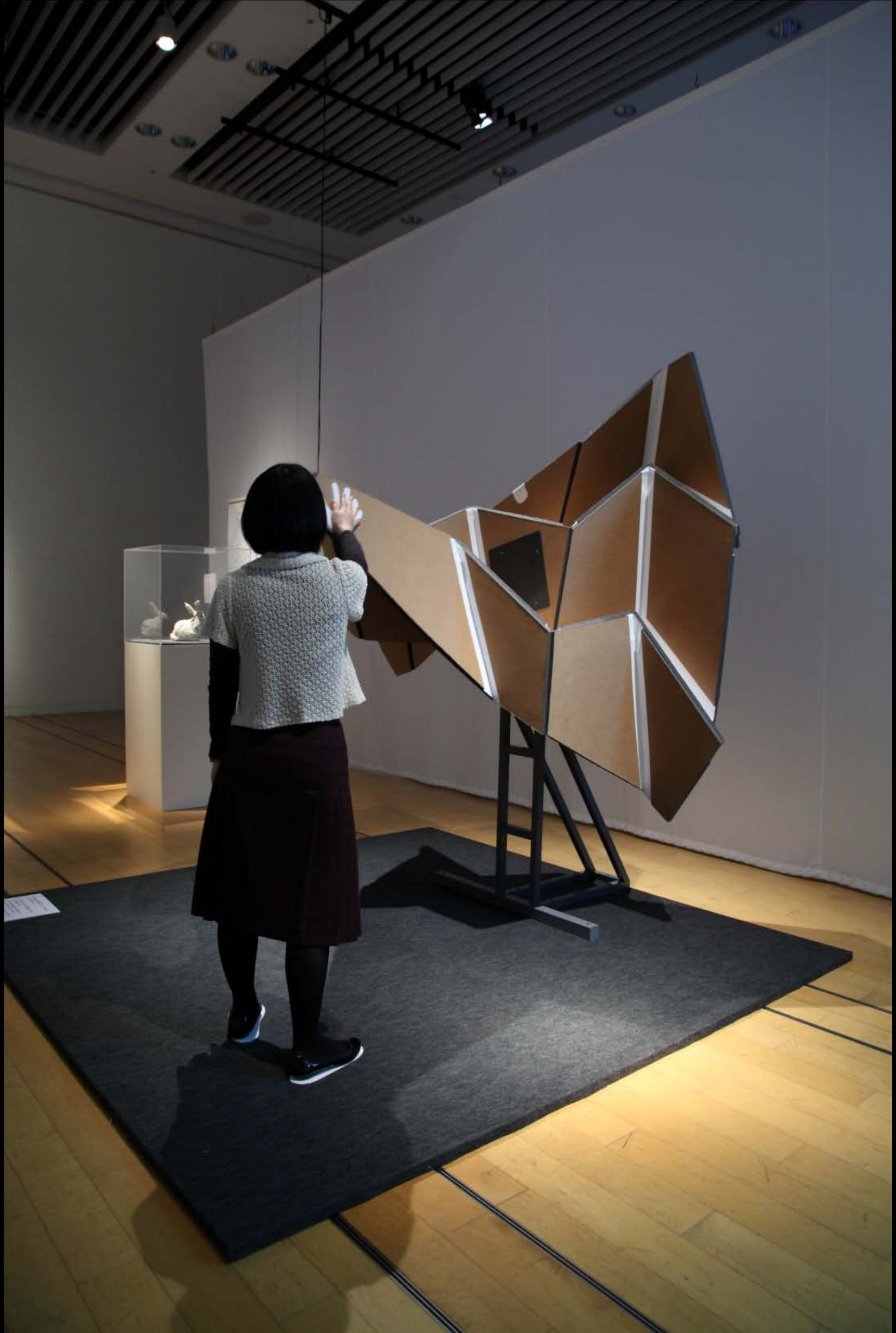


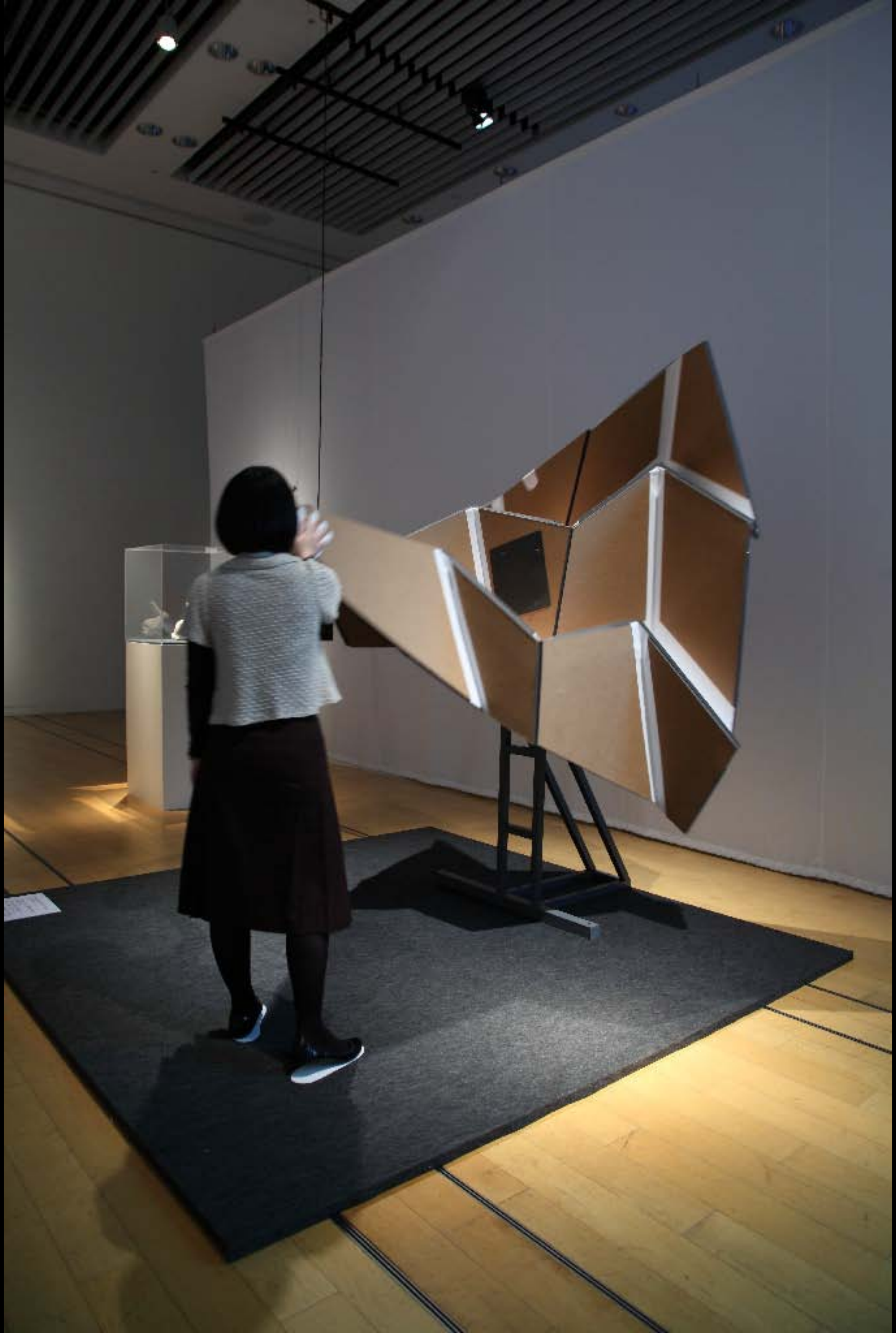
- Constant Thickness Model
  - the shape is locally defined
  - cf: Slidable Hinge →




miura-dome.3dm - Rhinoceros (教育版) - [Perspective]
Pick Y direction:
ｺﾏﾝﾄﾞ: _Delete
ｺﾏﾝﾄﾞ: Grasshopper
ﾒｯｼｭを作成しています... ｷｬﾝｾﾙするにはEscを押してください。
ｺﾏﾝﾄﾞ:
Perspective
Grasshopper - thickening+folding-no-taper-layout
File Edit View Arrange Solution Window Help
Params Logic Scalar Vector Curve Surface Mesh Intersect XForm Complex
Geometry
Primitive
Special
mesh_orig
rho
rho0
center
out
Thickness
10.0000
mesh_in
mesh_folded
mesh_unfold
thick
Autosave complete (28 seconds ago)
ScriptEditor
Utility functions
private void RunScript(OnMesh mesh_in, OnMesh mesh_folded, OnMesh mesh_unfold, double thic
{
int num_face = mesh_in.m_F.Count();
int num_vert = mesh_in.m_V.Count();
//OnMesh[] meshlist = new OnMesh[num_face];
OnBrep[] breplist = new OnBrep[num_face * 2];
List<int>[] vert2face = new List<int>[num_vert];
On3dVector[] normface = new On3dVector[num_face];
On3dVector[] normface_folded = new On3dVector[num_face];
On3dVector[] normface_unfold = new On3dVector[num_face];
for (int v = 0; v < num_vert; ++v) {
vert2face[v] = new List<int>();
}
for (int f = 0; f < num_face; ++f) {
OnMeshFace face = mesh_in.m_F[f];
int num_vert_face = face.IsQuad() ? 4 : 3;
On3dVector[] p = new On3dVector[num_vert_face];
On3dVector[] p_f = new On3dVector[num_vert_face];
On3dVector[] p_u = new On3dVector[num_vert_face];
for (int v = 0; v < num_vert_face; ++v) {
vert2face[face.get_vi(v)].Add(f);
p[v] = new On3dVector(mesh_in.m_V[face.get_vi(v)].x, mesh_in.m_V[face.get_vi(v)].y,
p_f[v] = new On3dVector(mesh_folded.m_V[face.get_vi(v)].x, mesh_folded.m_V[face.get
作業平面 x 213.148 y 36.206 z 0.000 0.000 ﾃﾞﾌｫﾙﾄ ｽﾅｯﾌﾟ 直交ﾓｰﾄﾞ 平面ﾓｰﾄﾞ

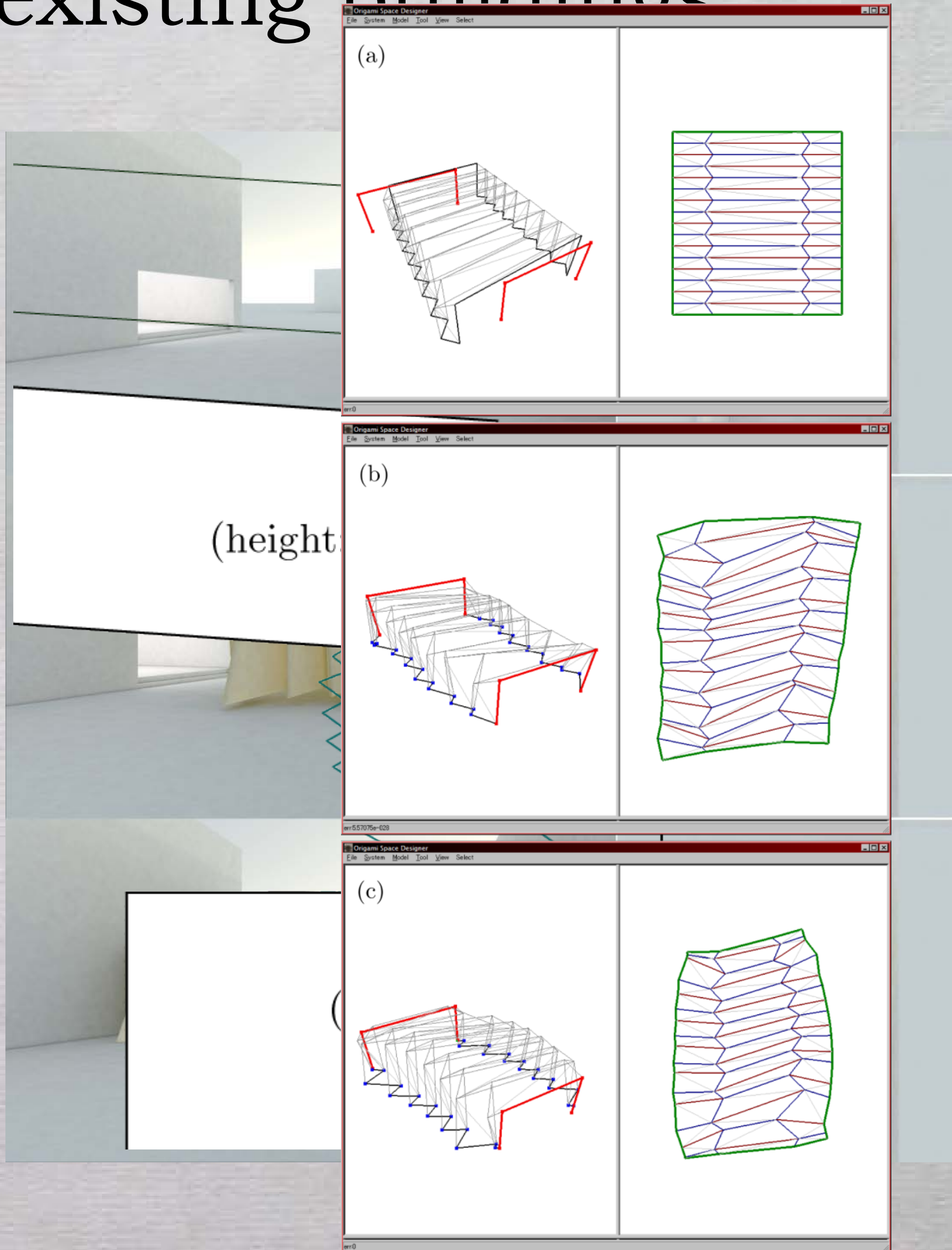






# Example: Construct a foldable structure that temporarily connects existing buildings

- Space: Flexible
  - Connects when opened
    - Openings: different position and orientation
    - Connected gallery space
  - Compactly folded
    - to fit the facade
- Structure: Rigid
  - Rigid panels and hinges



# Panel Layout