Computational Design of Reconfigurables

Akash Garg[1]
Alec Jacobson[1,2]
Eitan Grinspun[1]

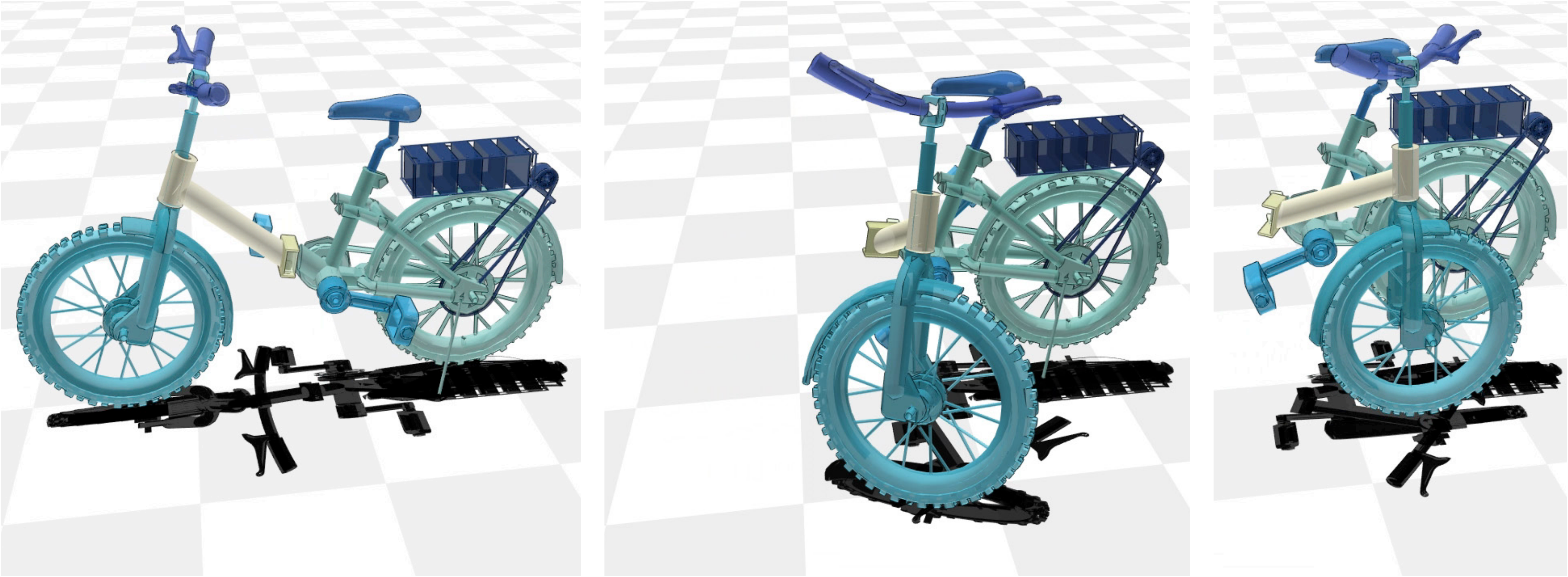
[1]*Columbia University*
[2]*University of Toronto*

reconfigurable | riːkənˈfɪɡ(ə)rəb(ə)l |

noun

an object or collection of objects whose transformation between various states defines its functionality or aesthetic appeal.

Foldable bicycle is a classic example of a reconfigurable



state #1 *transition* state #2

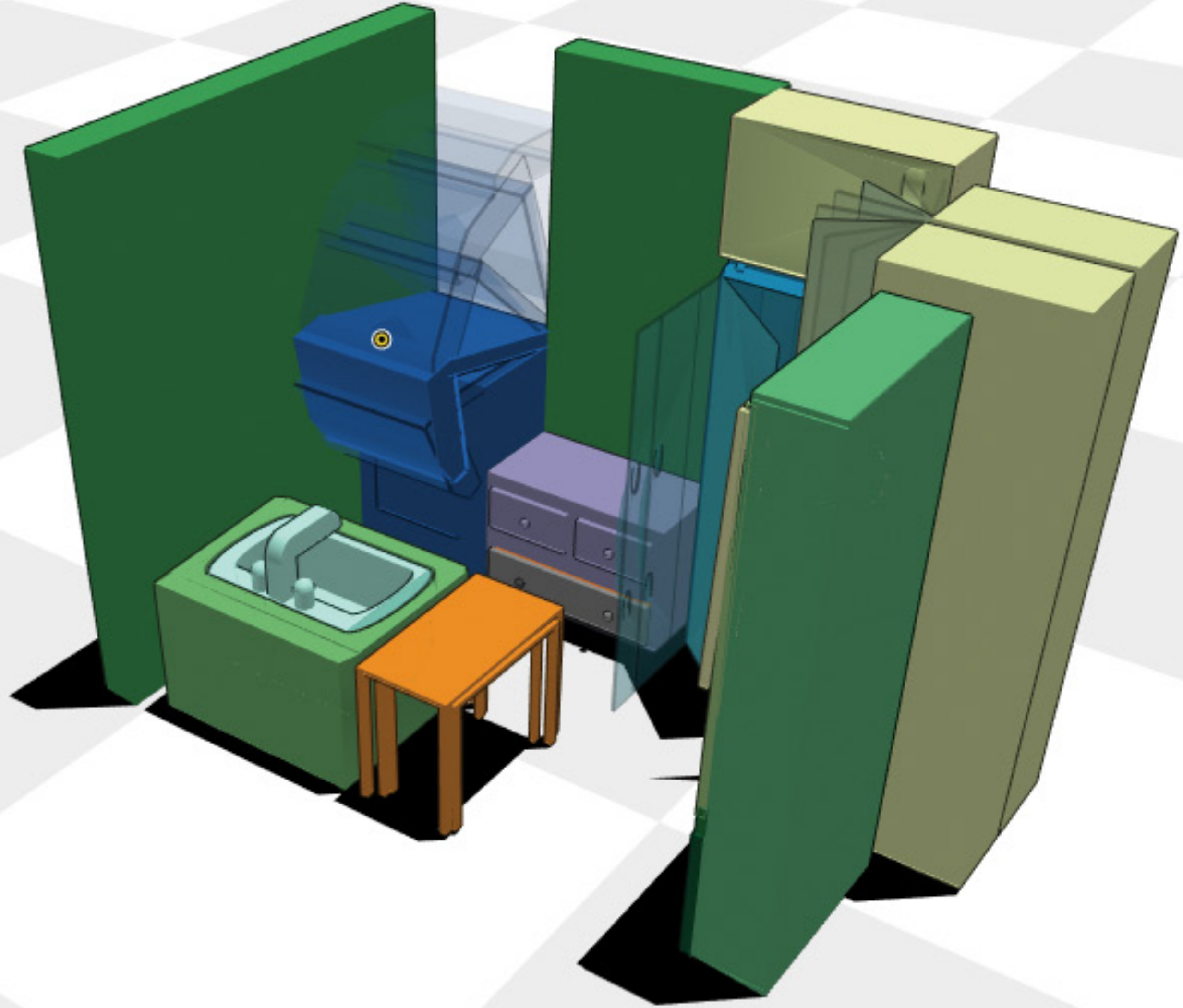
Transforming furniture is also a reconfigurable

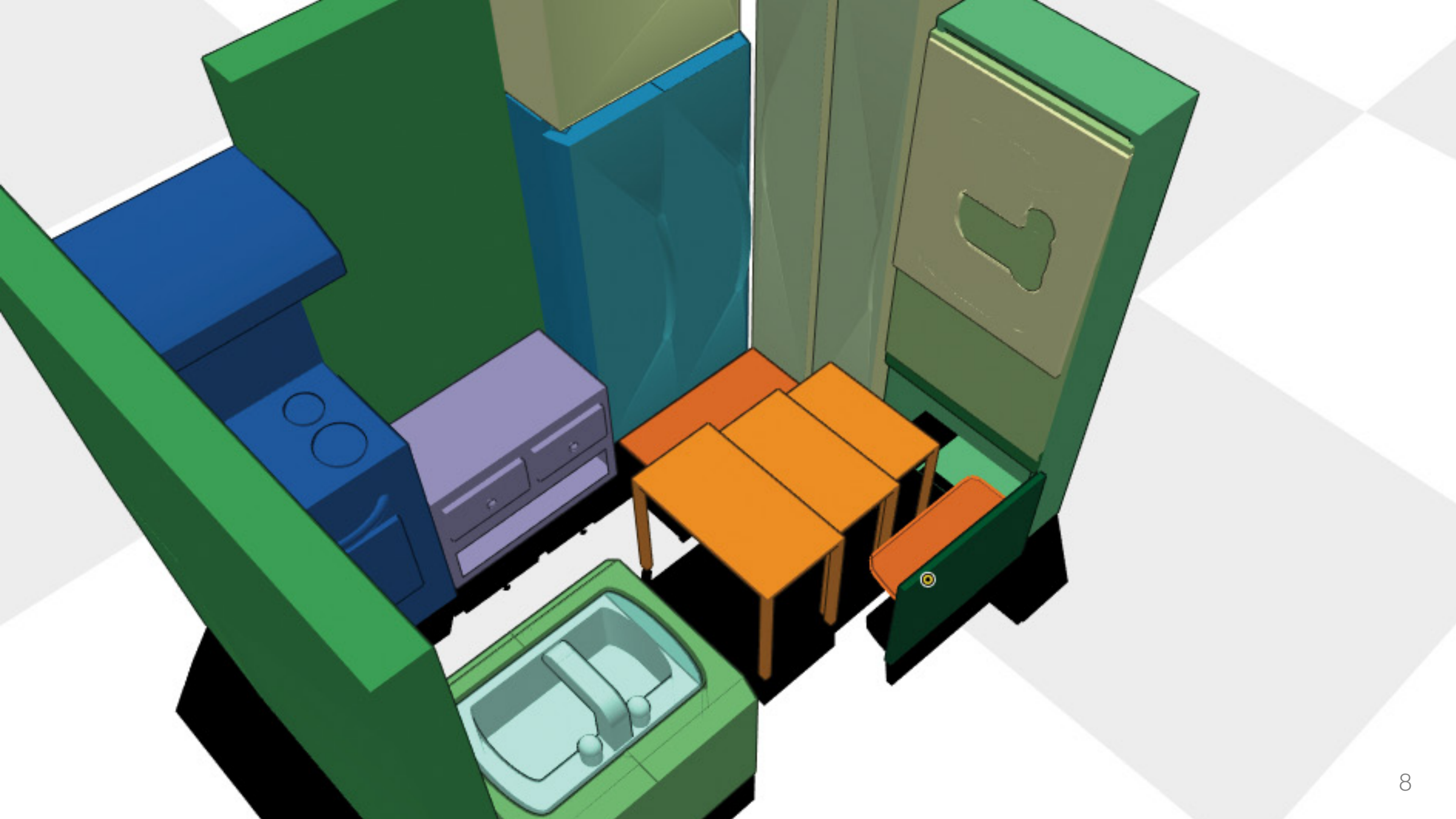
https://youtu.be/A3PDiDLiOEI

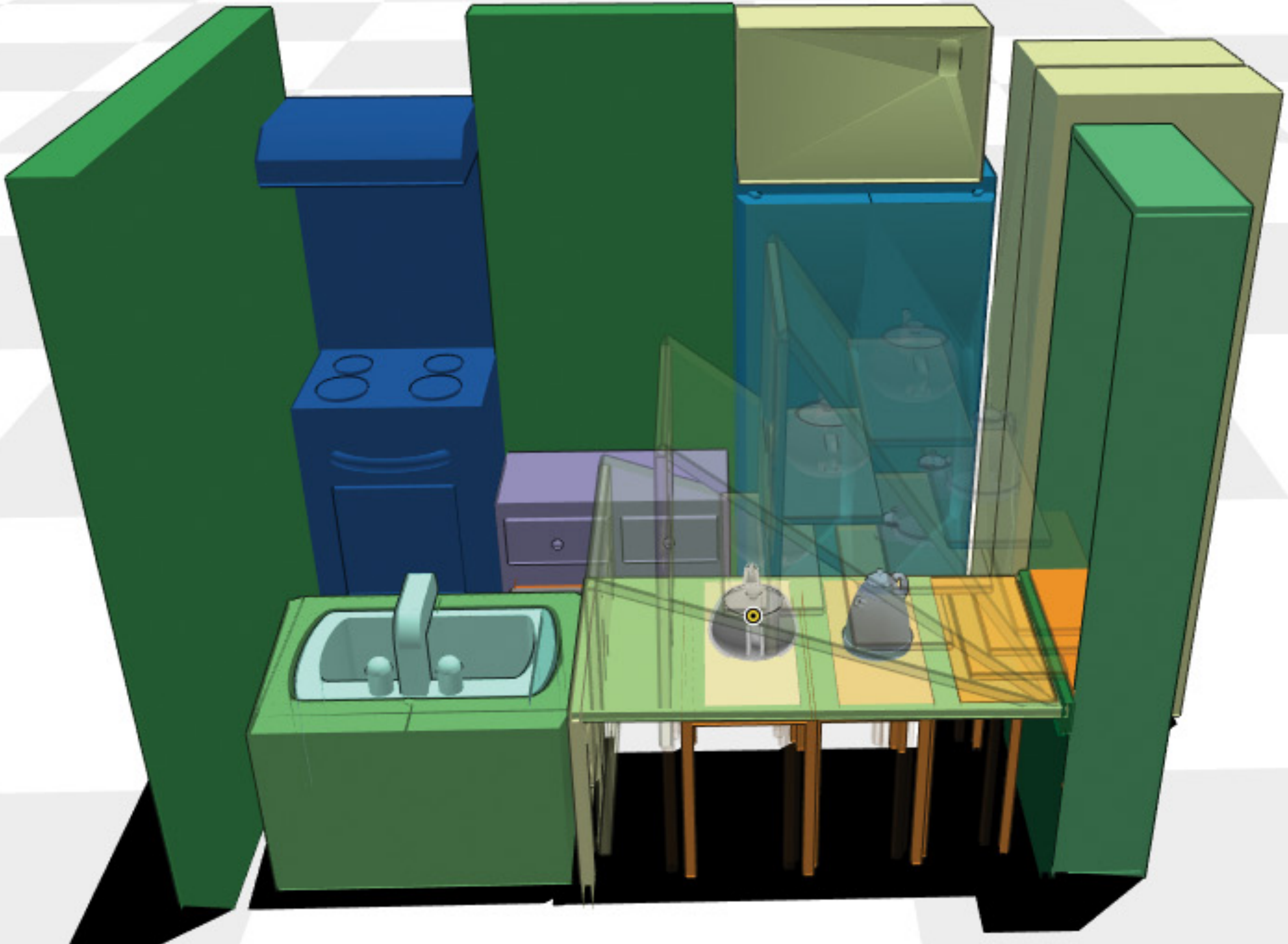
Transforming furniture is also a reconfigurable

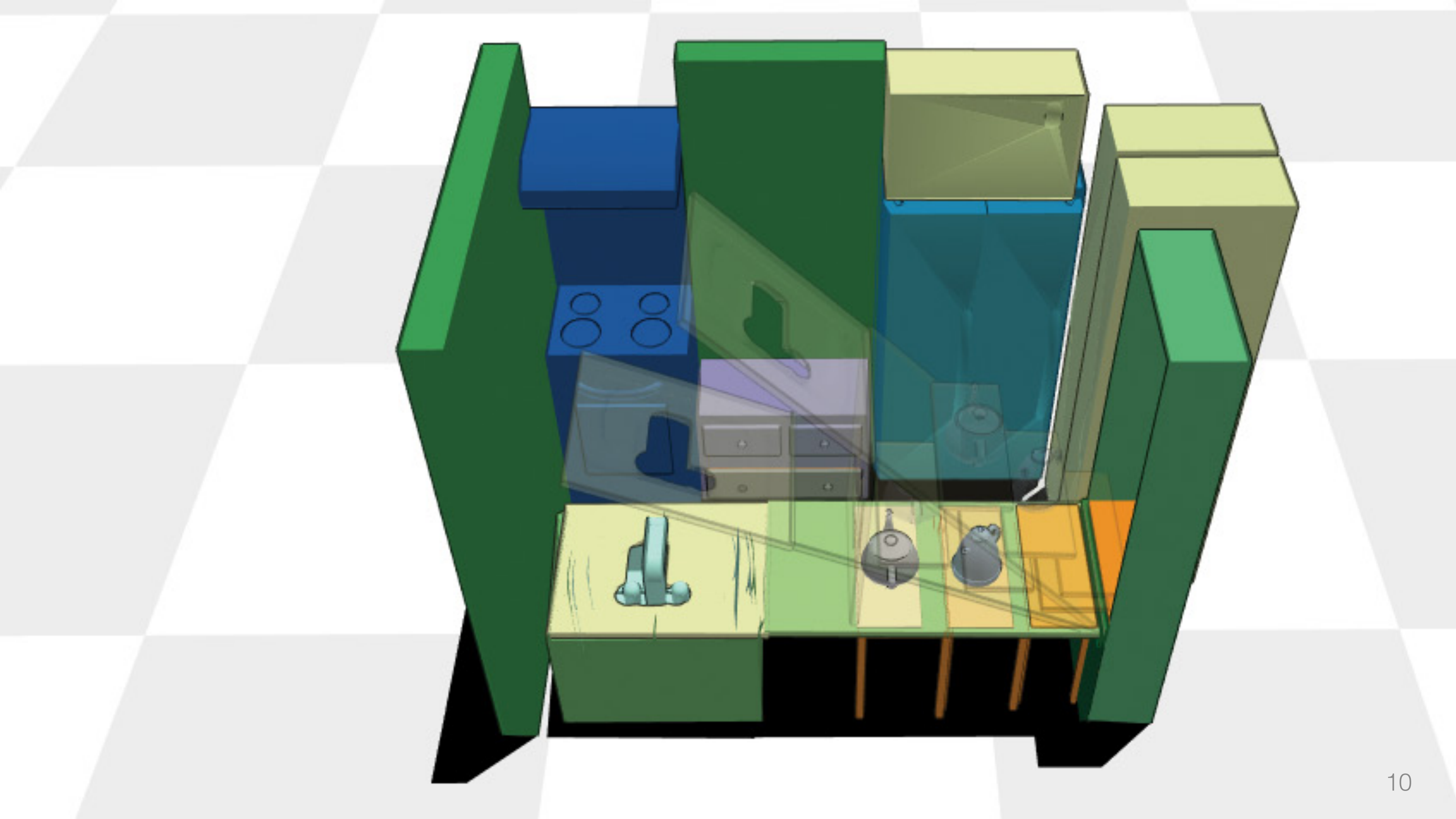
https://youtu.be/A3PDiDLiOEI

Space-saving kitchen is also a reconfigurable

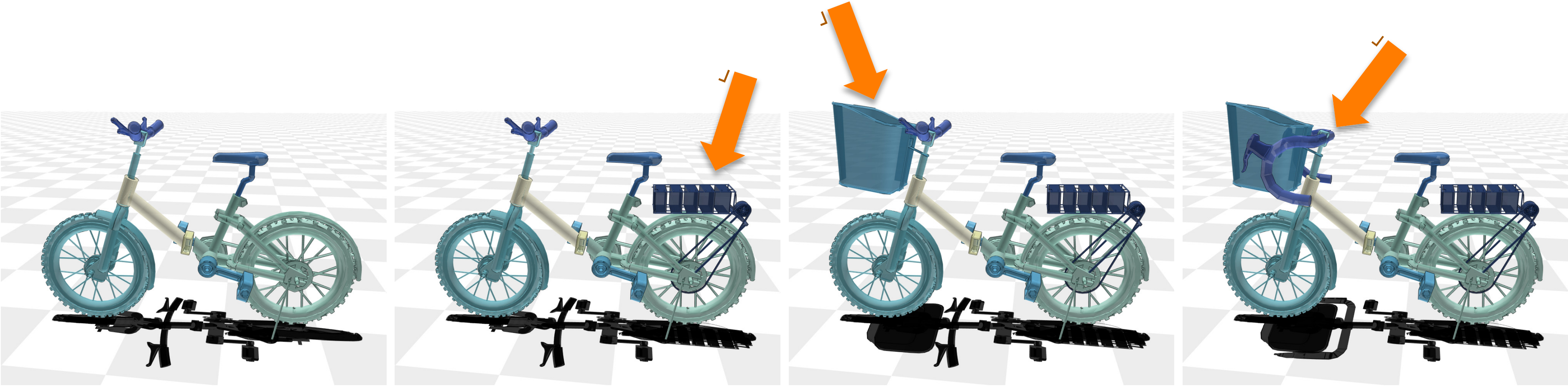






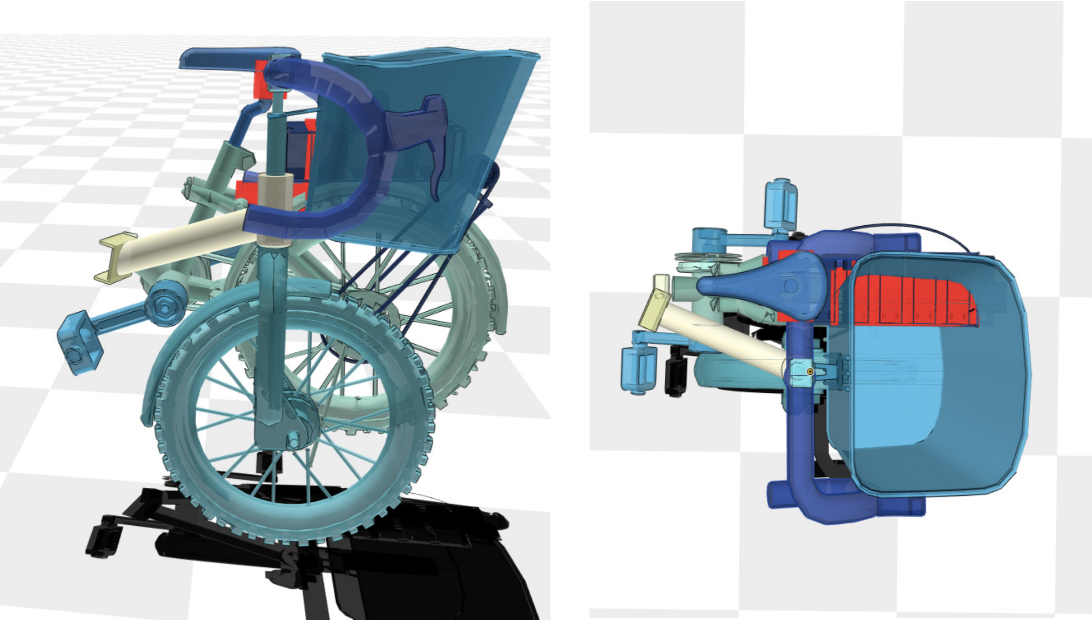


Problems occur if we design each state in isolation…



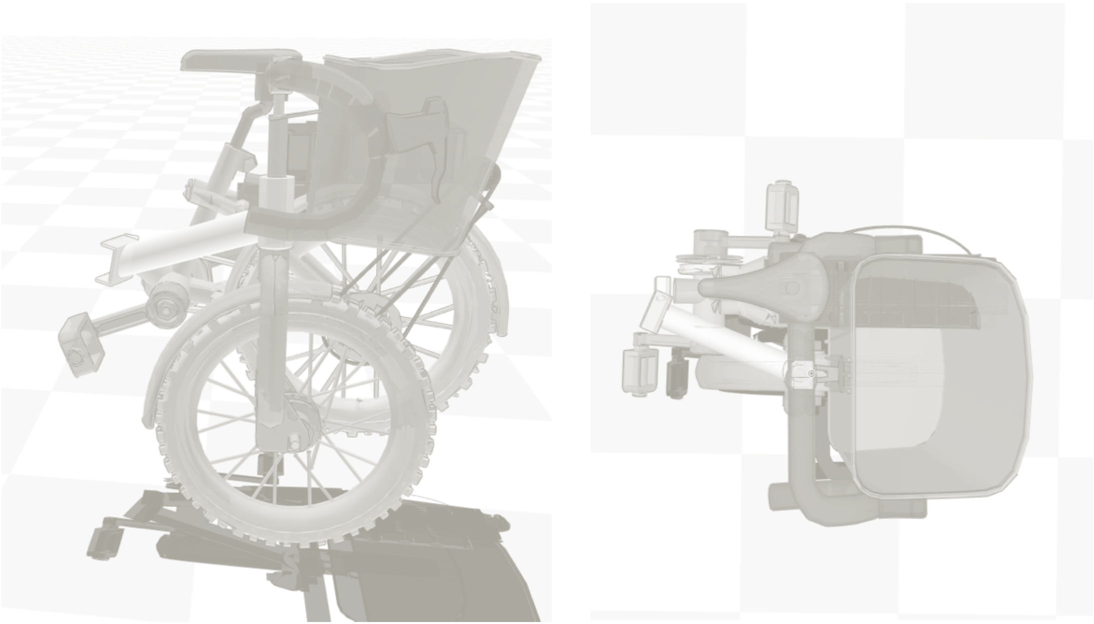
state #1

...collisions prevent transitioning to other states

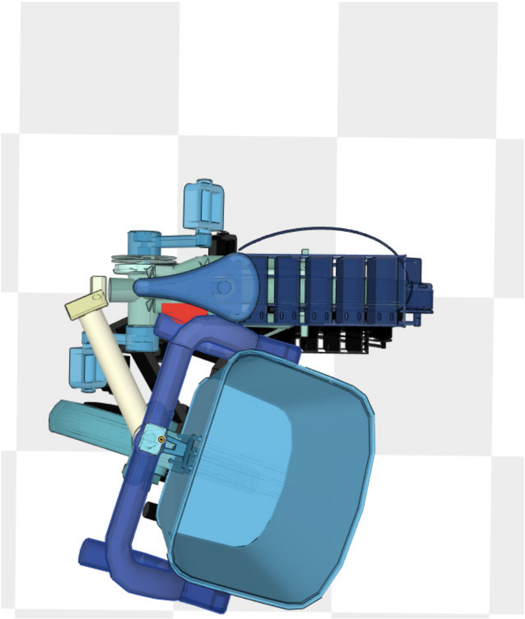


state #2

...collisions prevent transitioning to other states

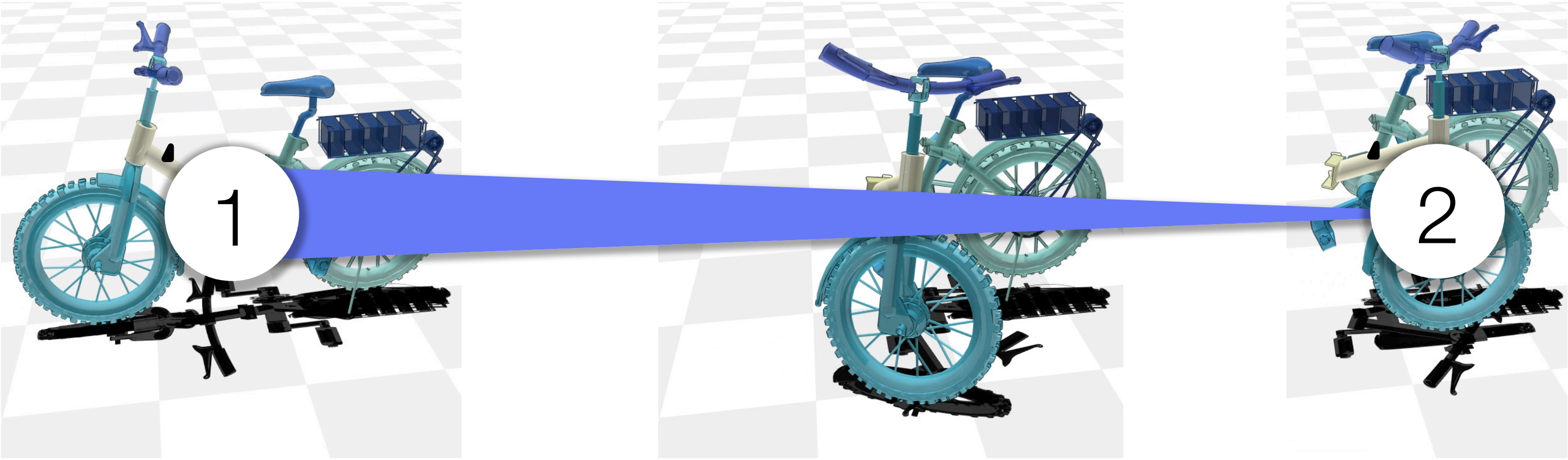


state #2

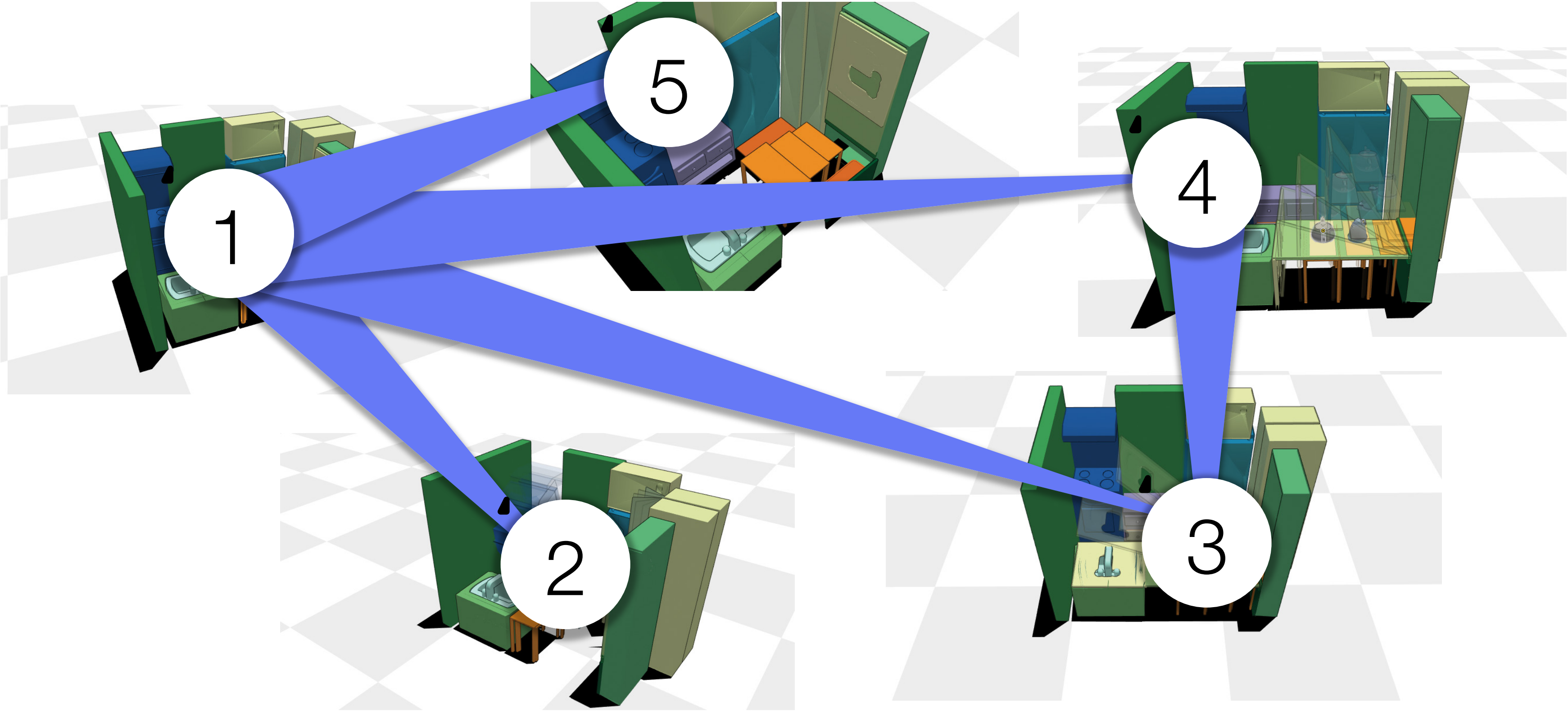


first contact

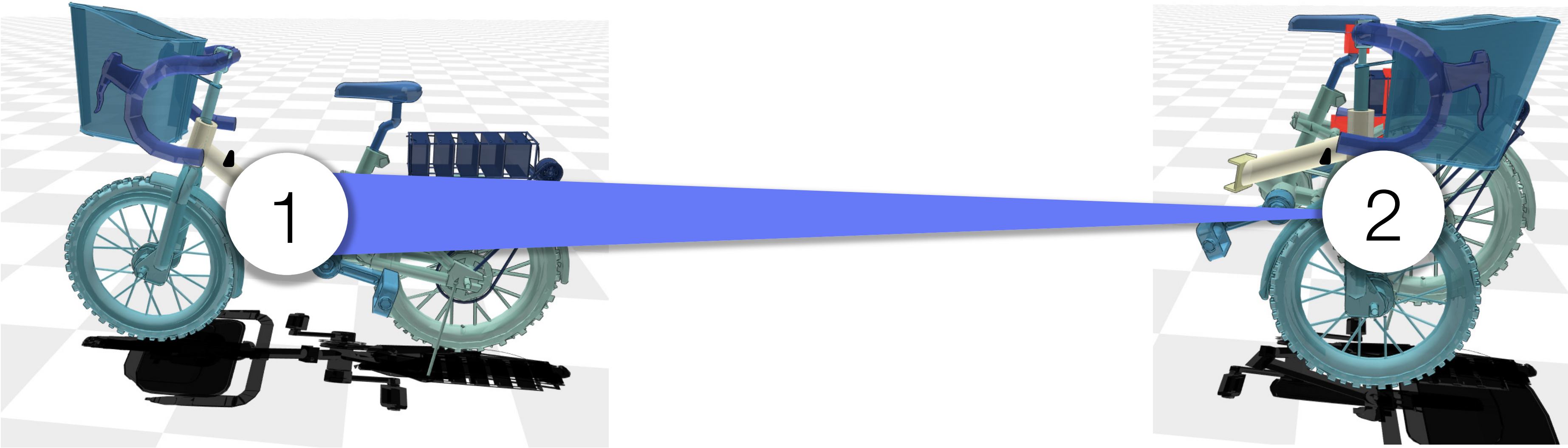
We model reconfigurables as a graph of transitions



We model reconfigurables as a graph of transitions



Our goal is to help the designer keep all transitions collision free

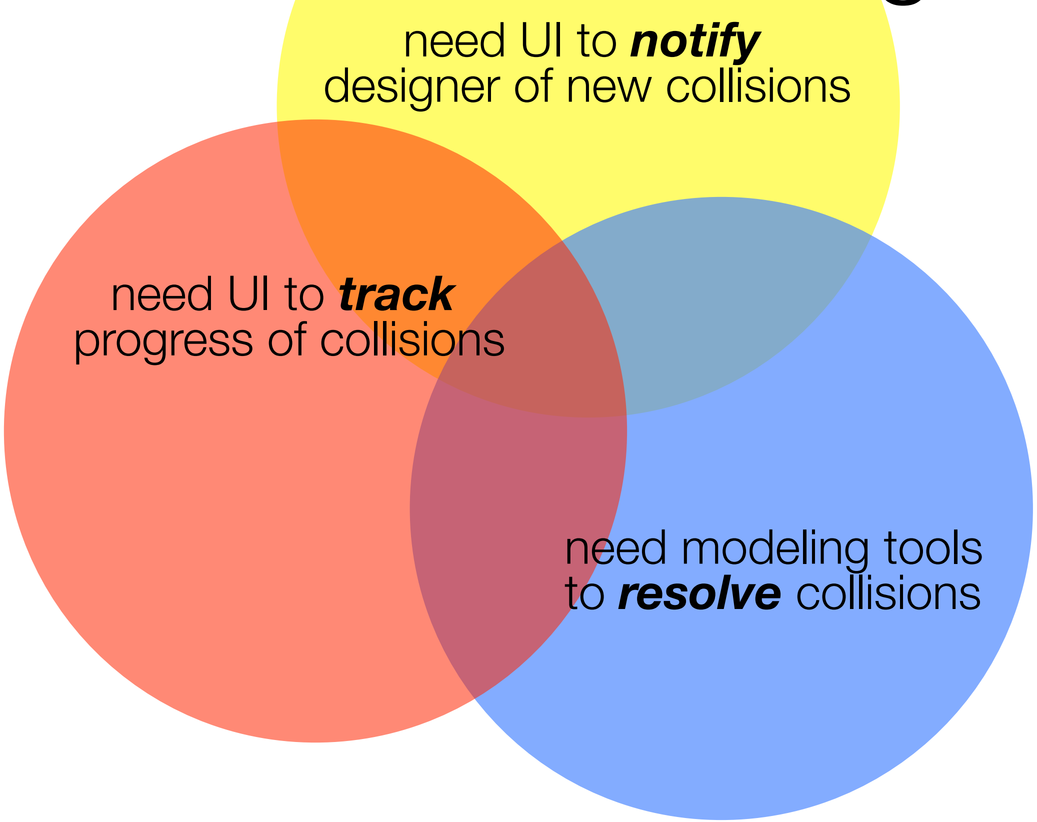


need UI to ***notify***
designer of new collisions

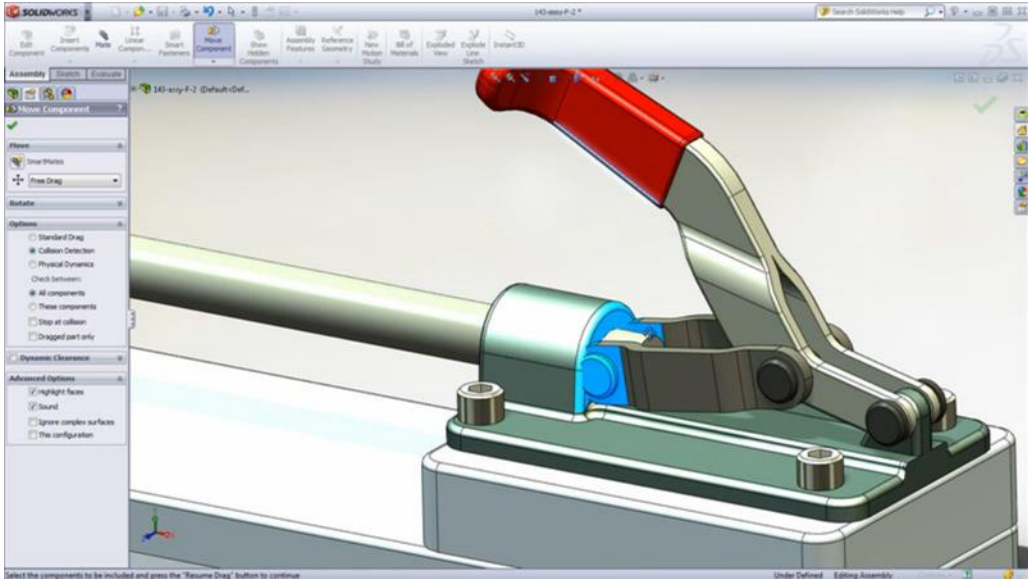
need UI to ***track***
progress of collisions

need modeling tools
to ***resolve*** collisions

Harmonization of all three enables to rapid and creative editing sessions



Traditional CAD tools have limited collision support



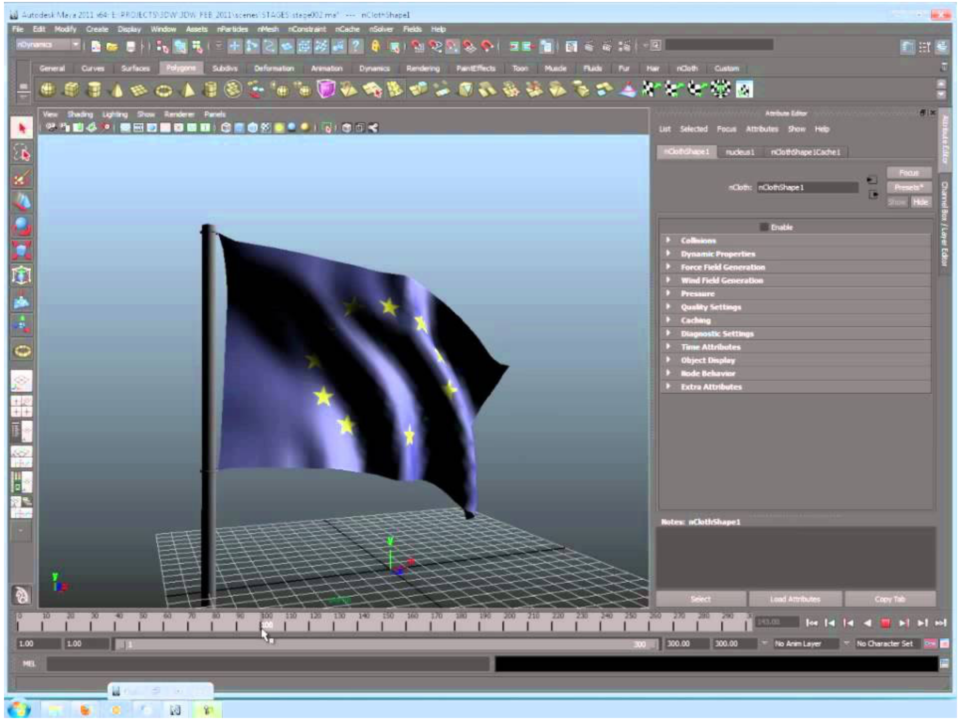
solidworks

offline detection

notification via “audible bell”

no automatic resolution

Animation tools detect and respond to collisions for physically-based effects



maya

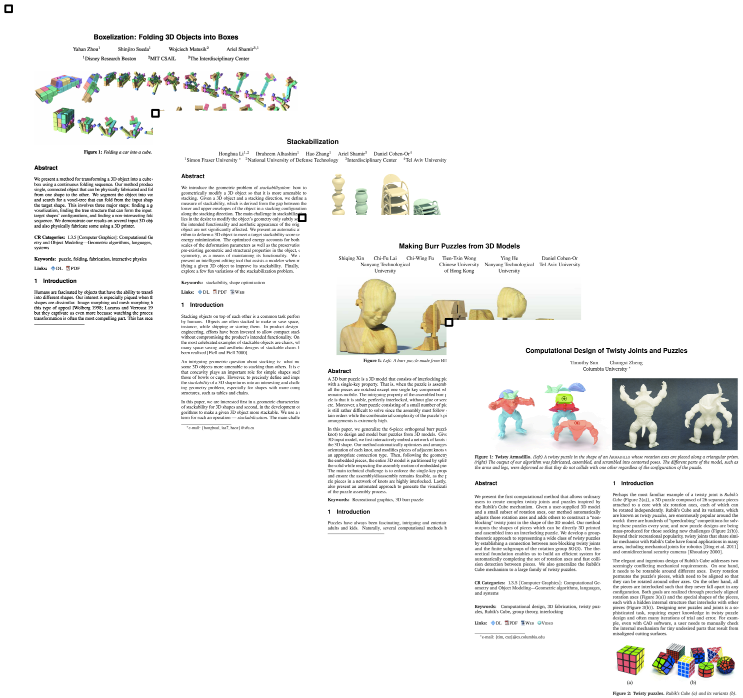
time is linear and “one-way”

no salient *states*

overlaps OK if not noticeable

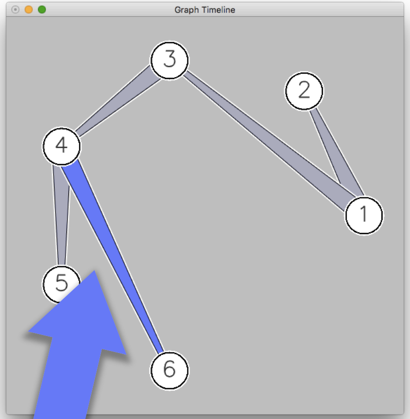
Previous computational design tools either:

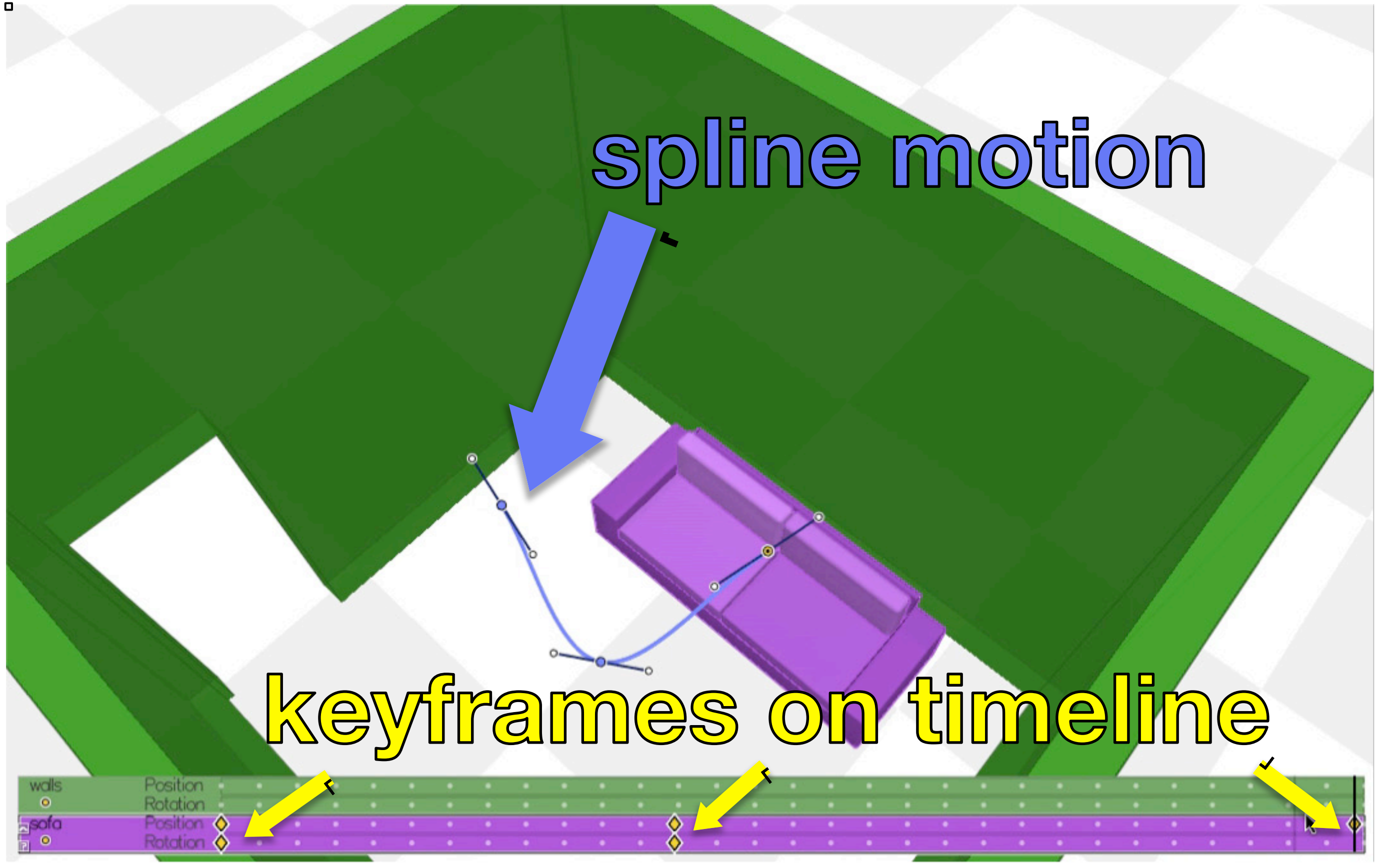
ignore collisions



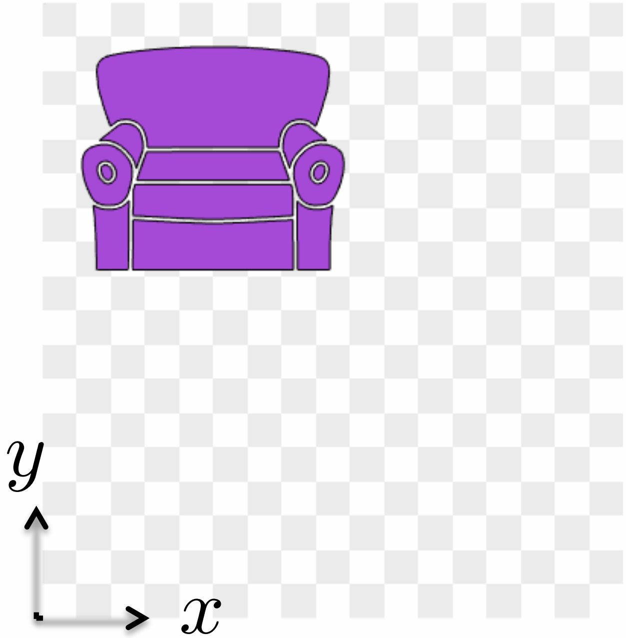
leverage a
specialized subspace

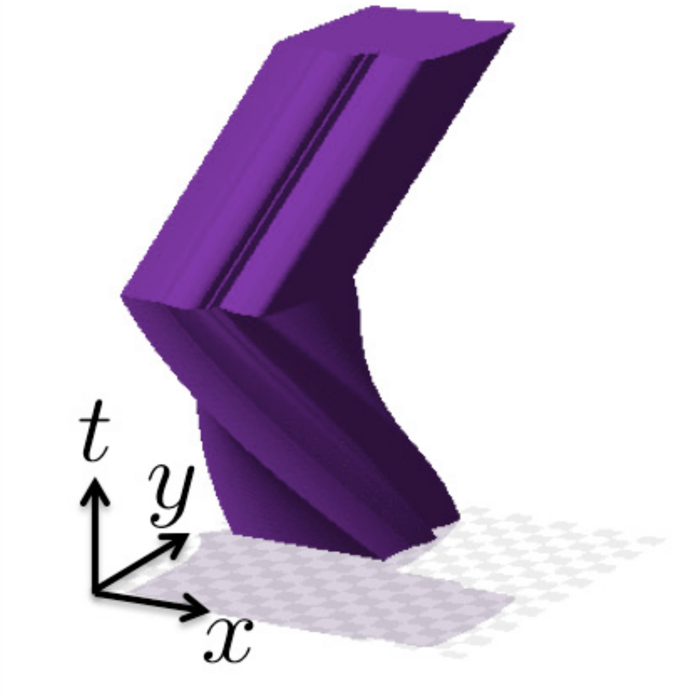
We begin with objects as rigid triangle meshes



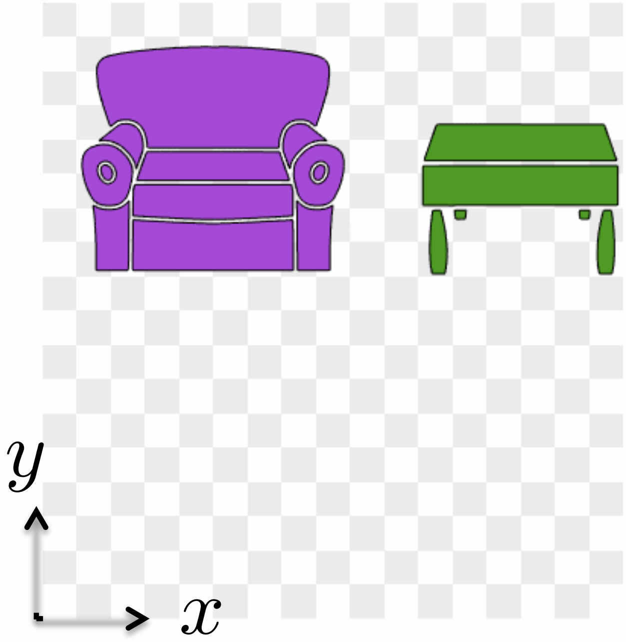


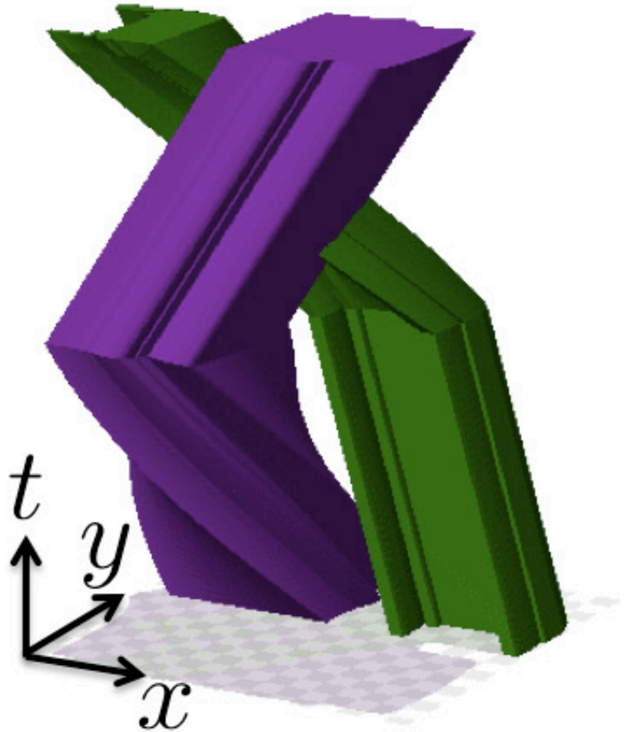
Along each transition an object is a 4D *spacetime* volume



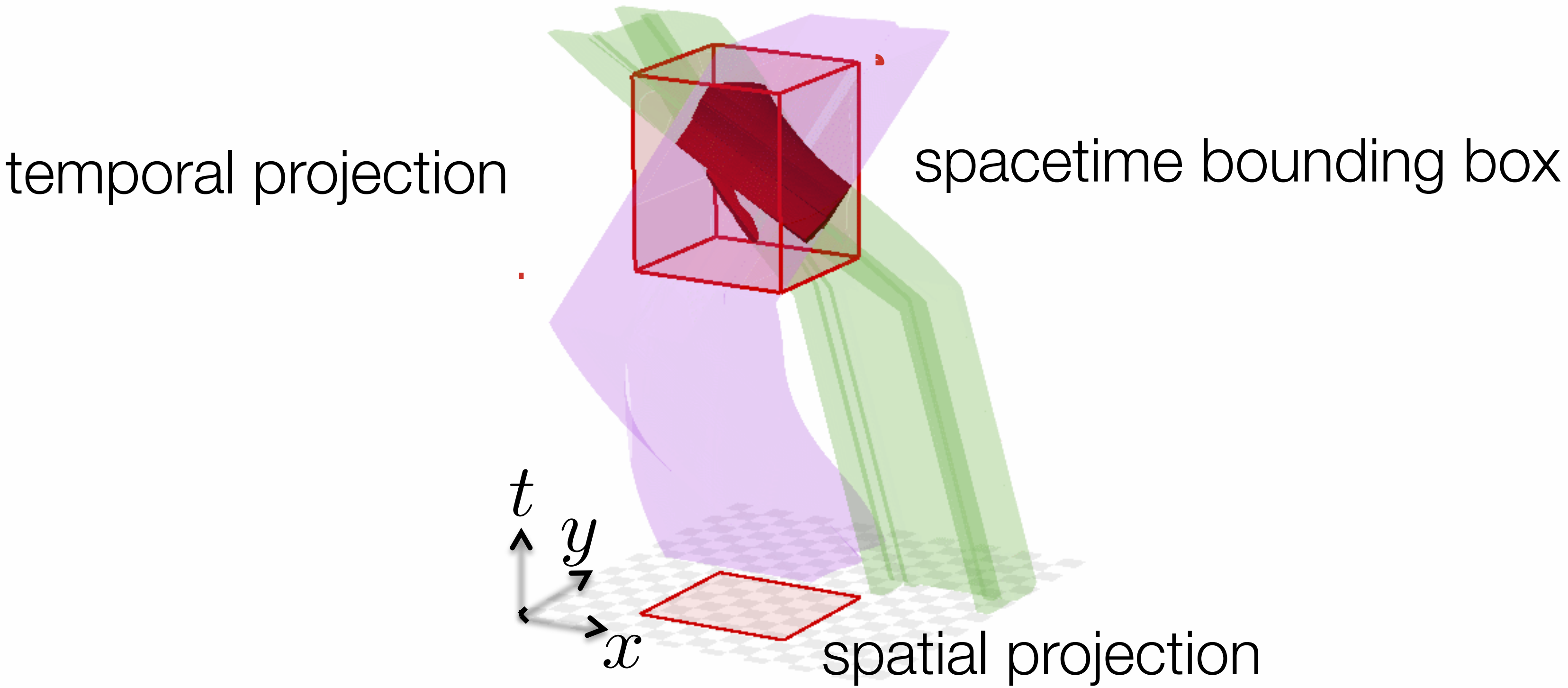


Collisions correspond to intersections in spacetime



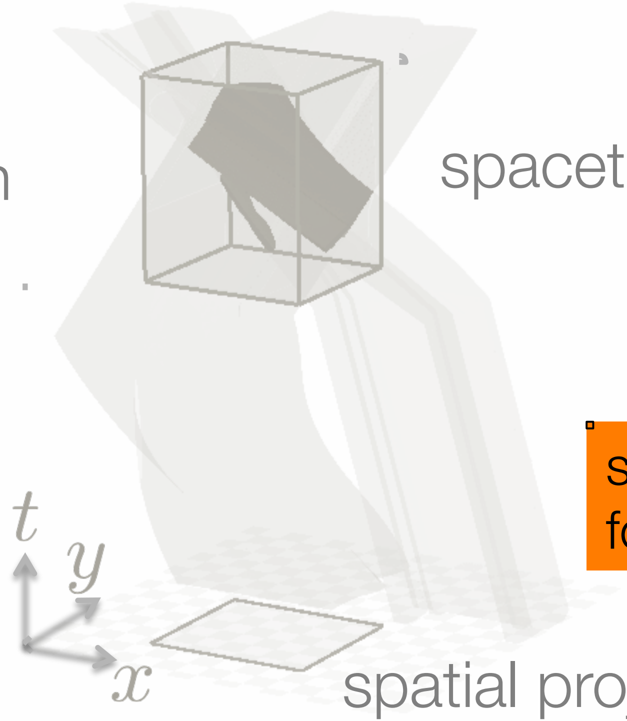


Collision Intervals form the foundation of our notification and tracking UIs



Collision Intervals form the foundation of our notification and tracking UIs

temporal projection

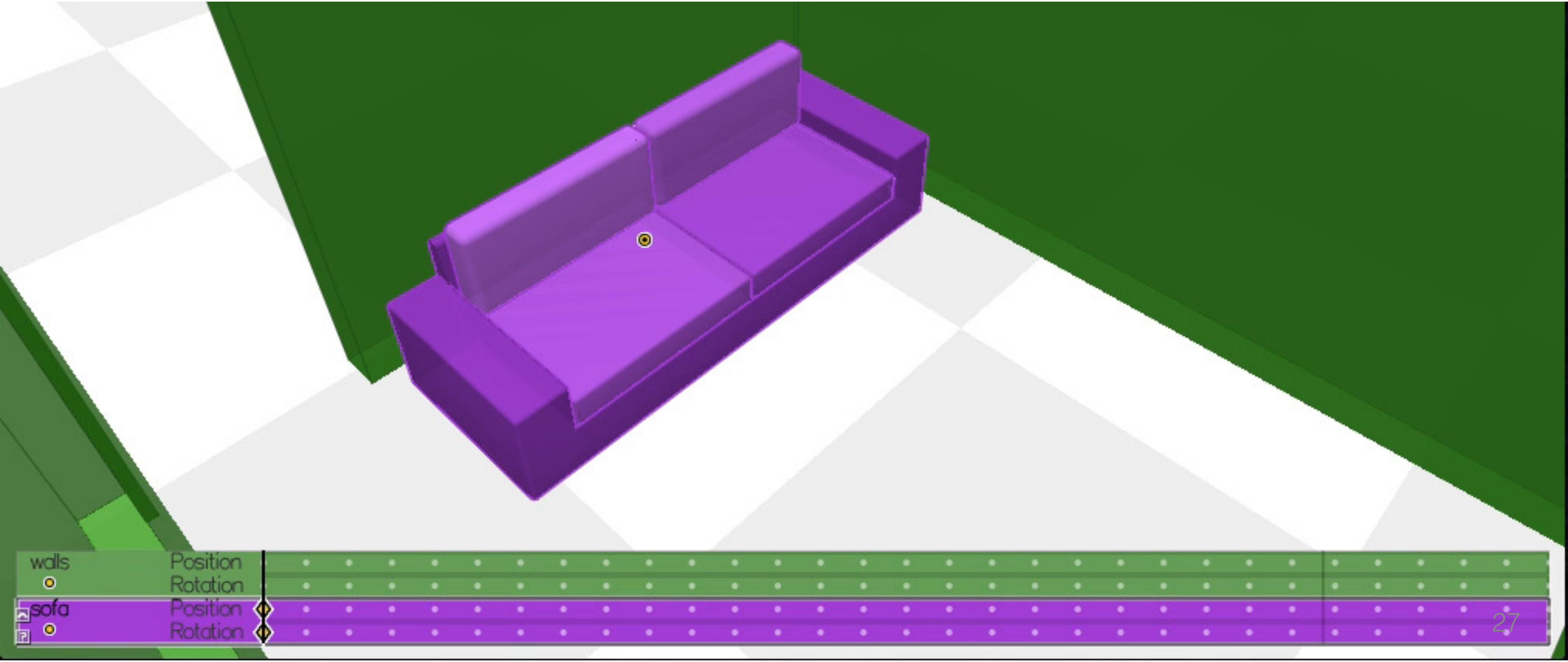


spacetime bounding box

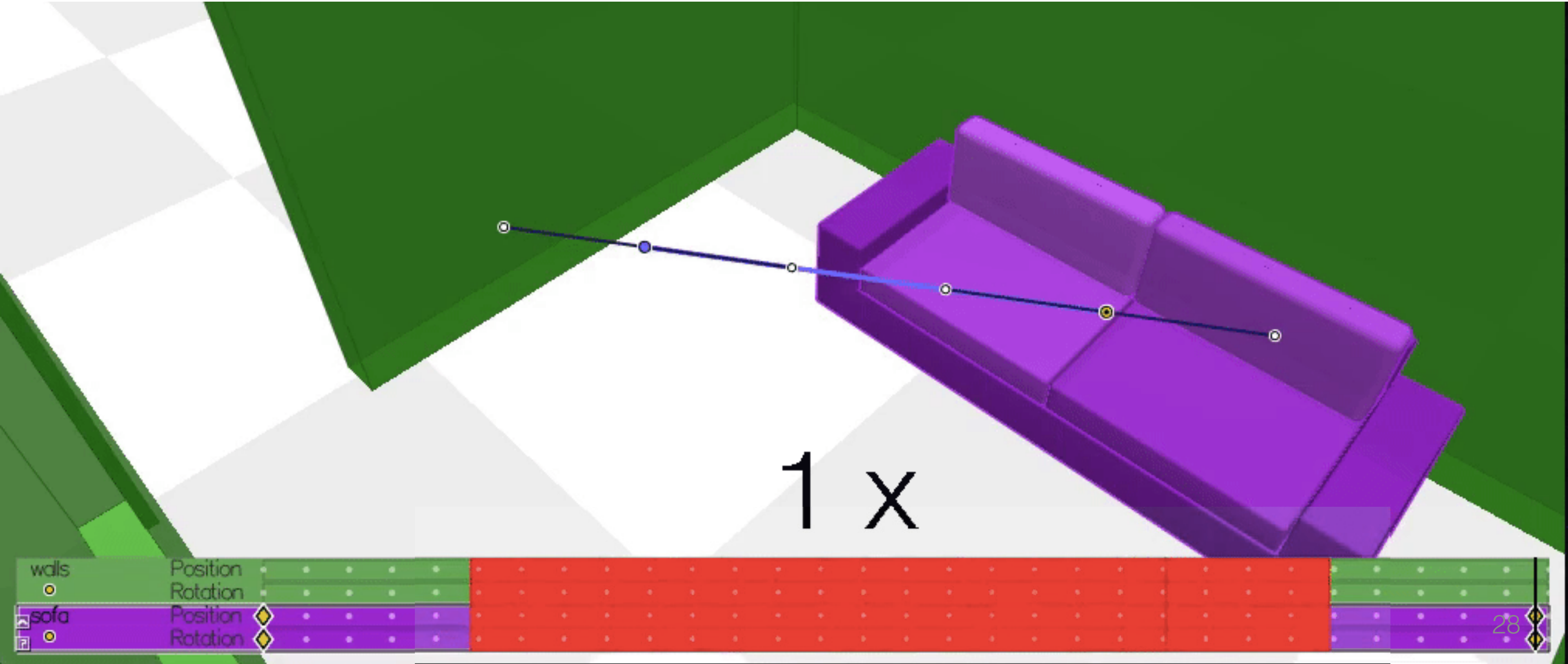
see paper about 4D *k*-DOP for collision interval detection

spatial projection

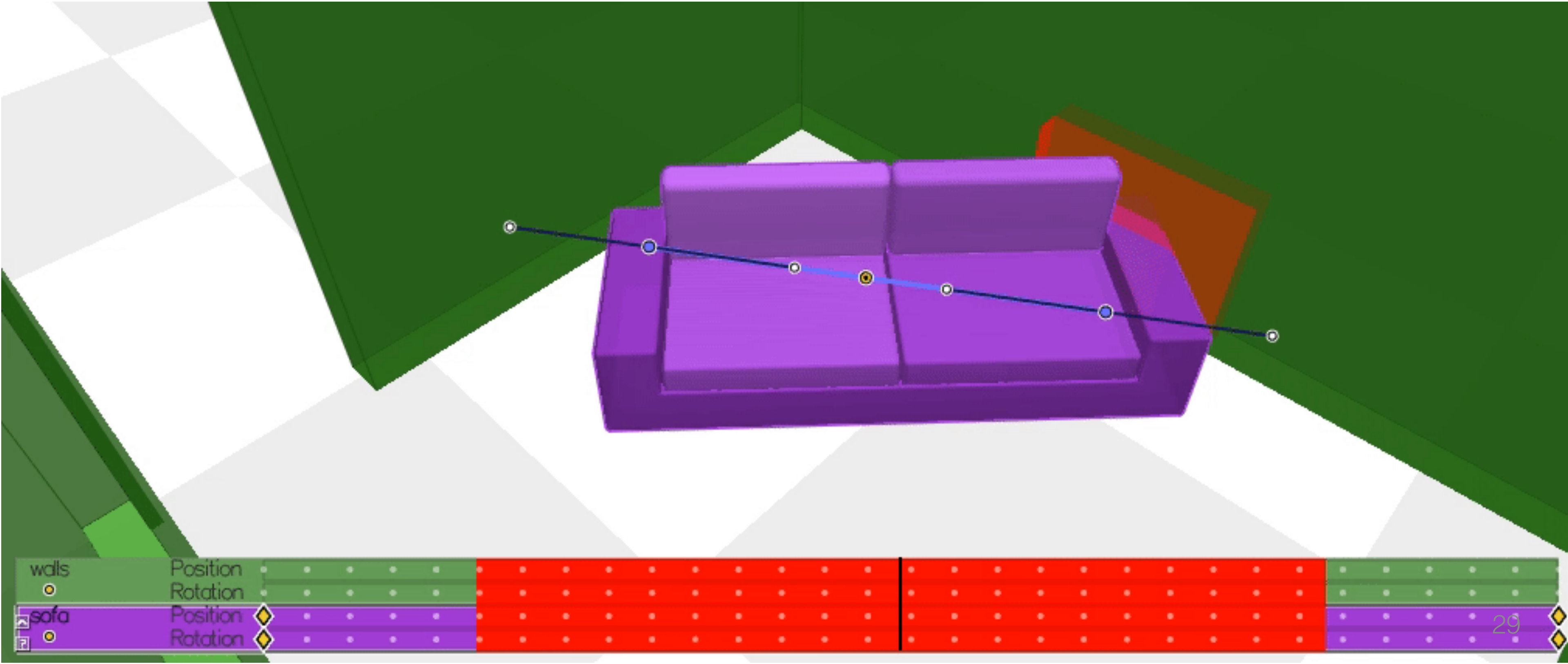
Spatial and temporal projections of collision intervals alert the designer



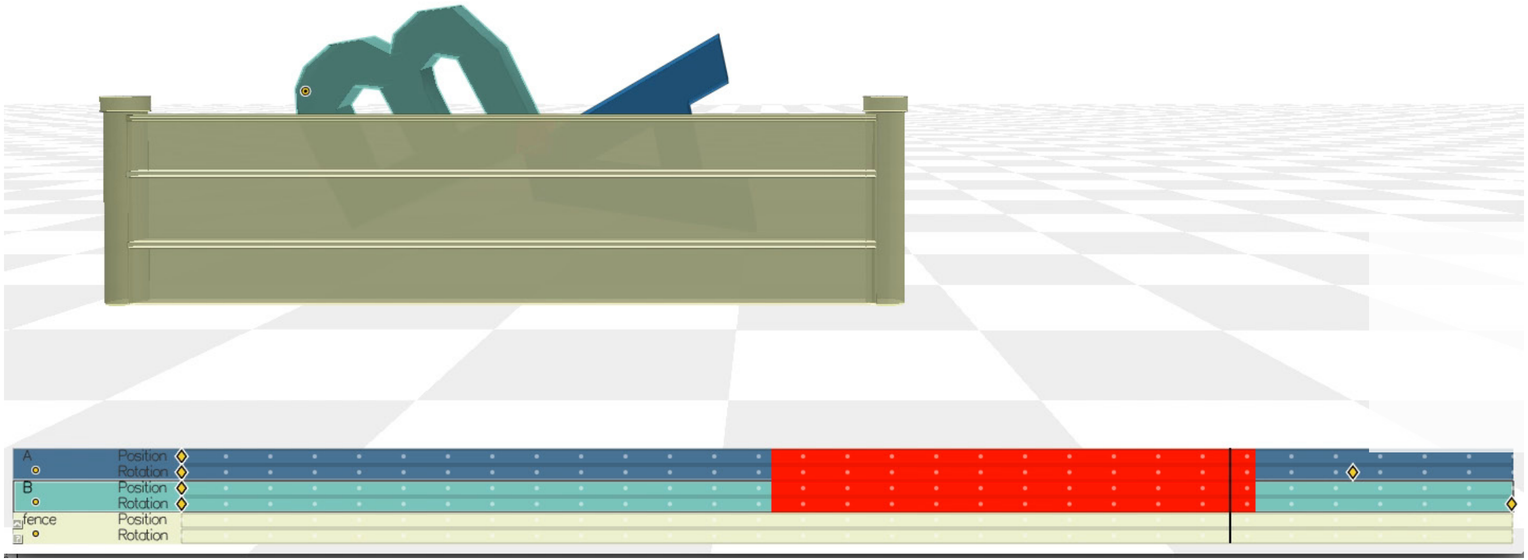
First priority: alert designer *when* collision is occurring



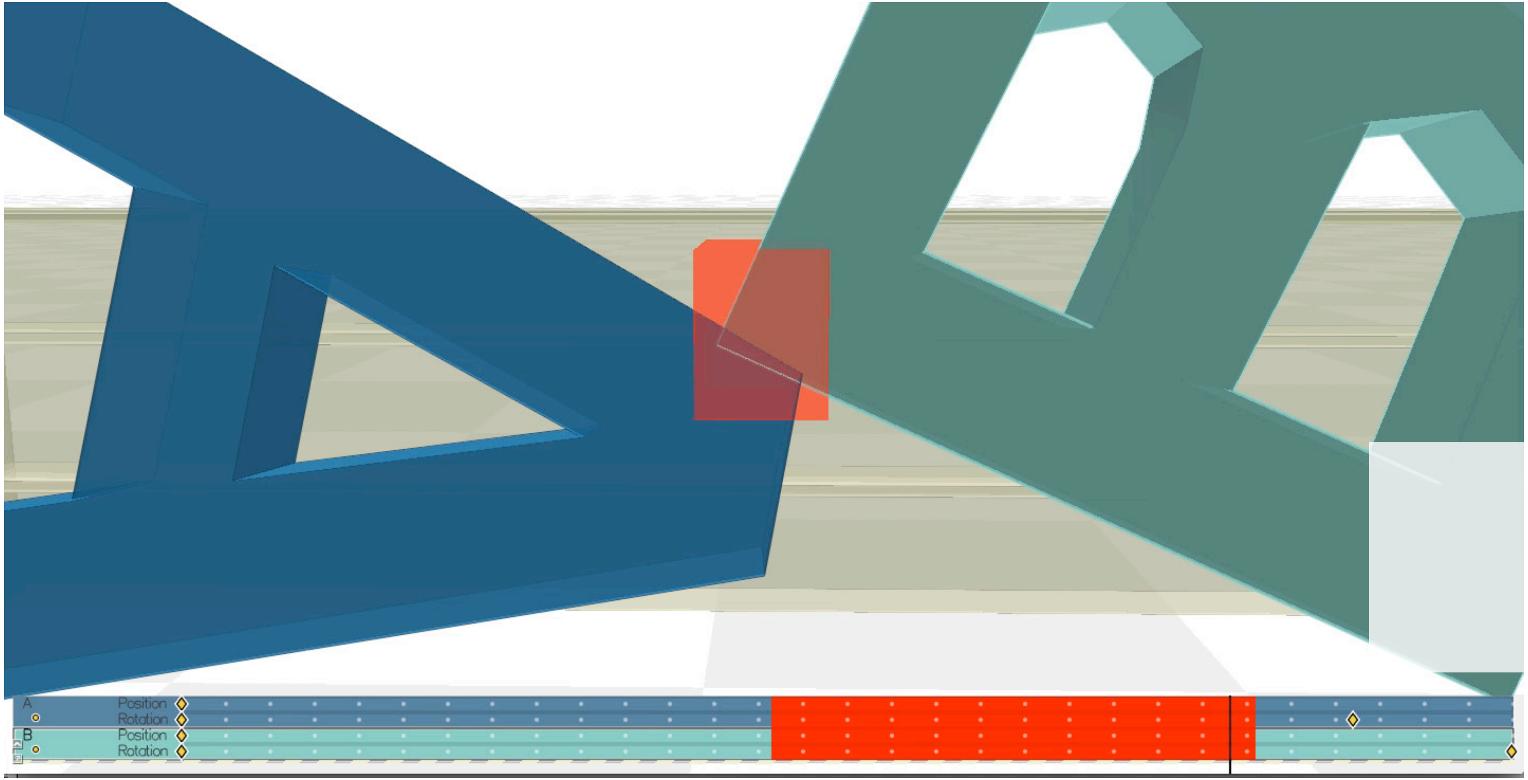
Next: alert designer *where* collision is occurring



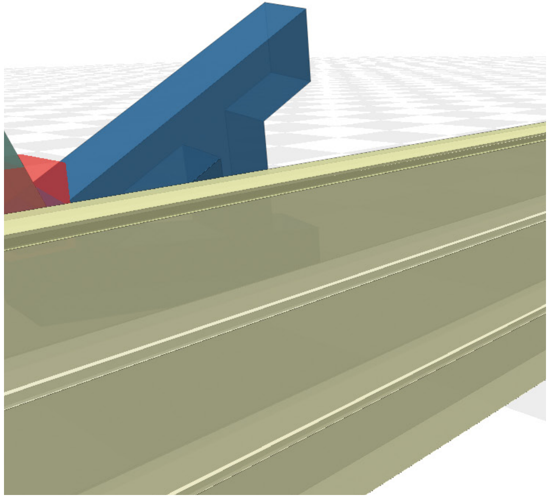
Designer’s current view may be poor, *hotkey* tumbles to optimized view



Designer's current view may be poor, *hotkey* tumbles to optimized view

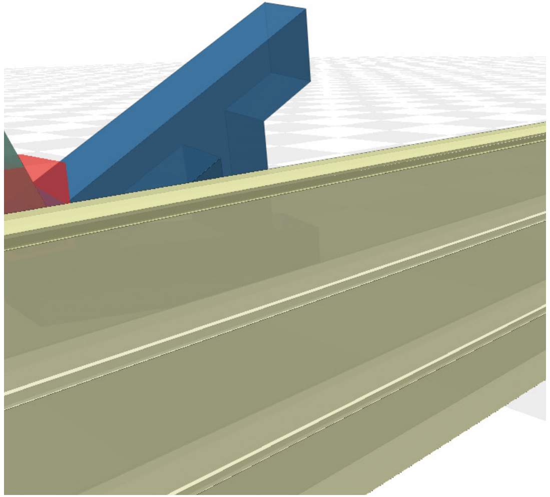


What makes a poor view?

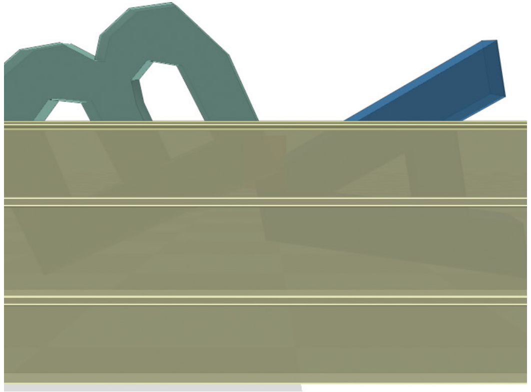


Collision interval
not central

What makes a poor view?

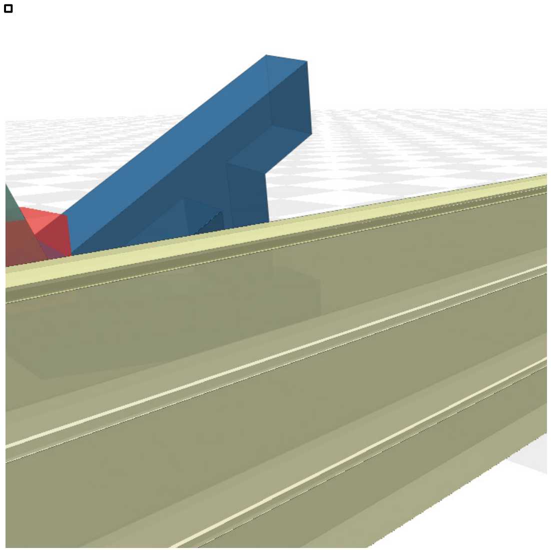


Collision interval
not central

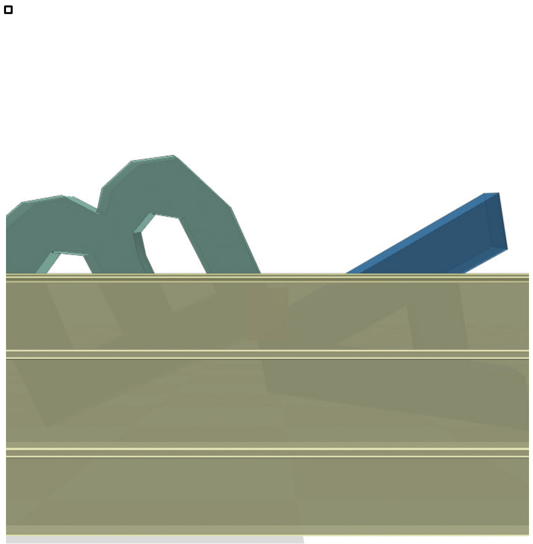


Collision interval
occluded

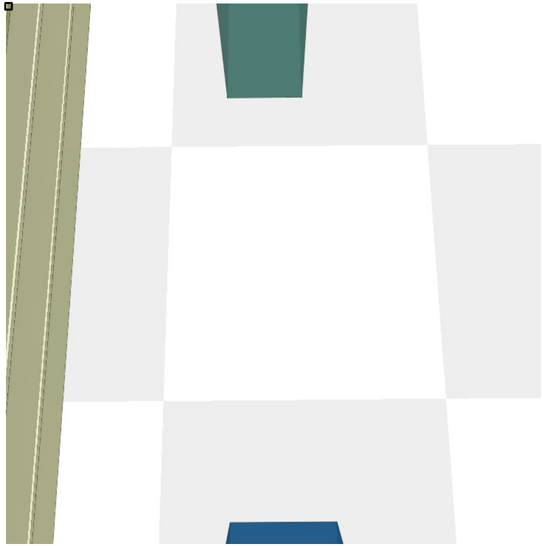
What makes a poor view?



Collision interval
not central

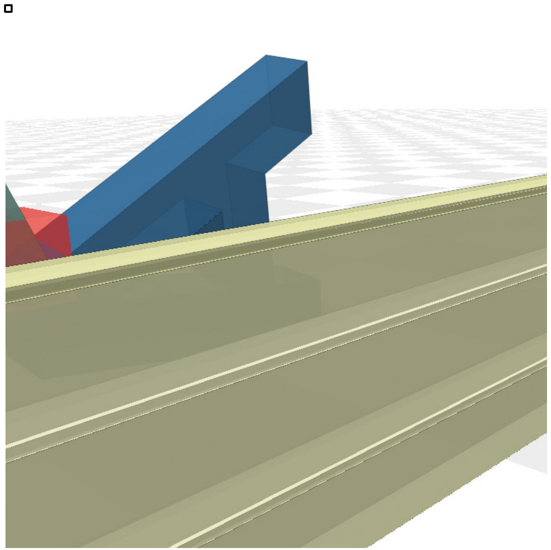


Collision interval
occluded

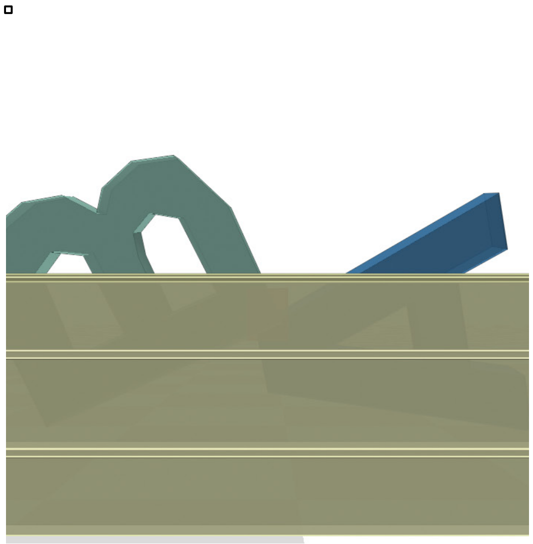


View direction
along trajectory

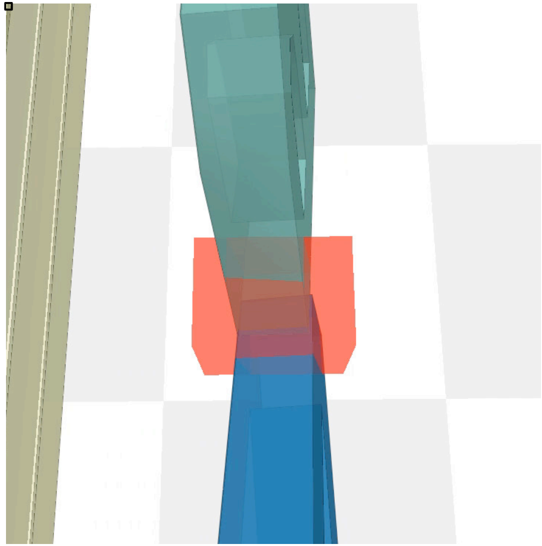
What makes a poor view?



Collision interval not central

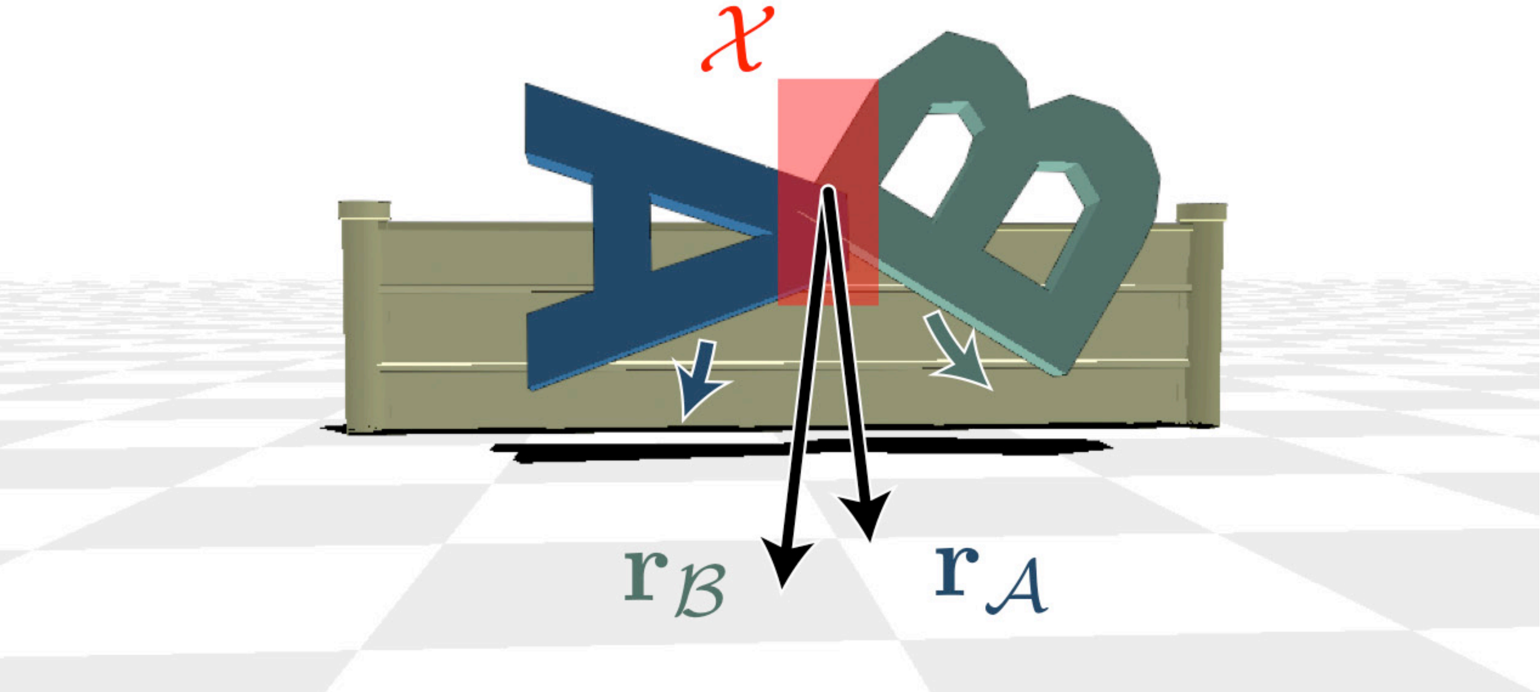


Collision interval occluded



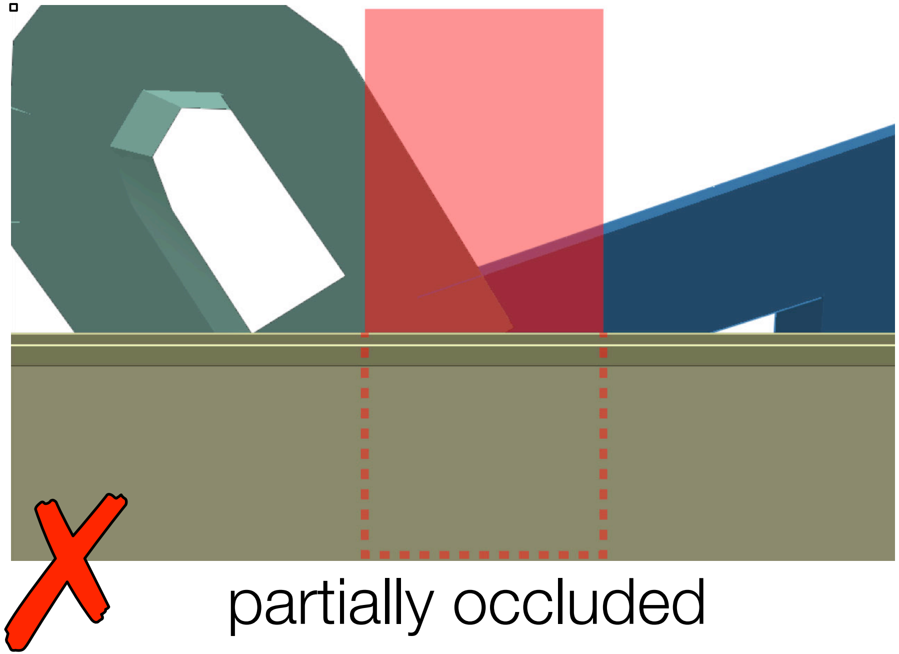
View direction along trajectory

For two objects, center collision interval along view direction orthogonal to motions…



Generally, two choices left: pick direction with least occlusion

$$\mathbf{r}_{\mathcal{A}} \times \mathbf{r}_{\mathcal{B}}$$

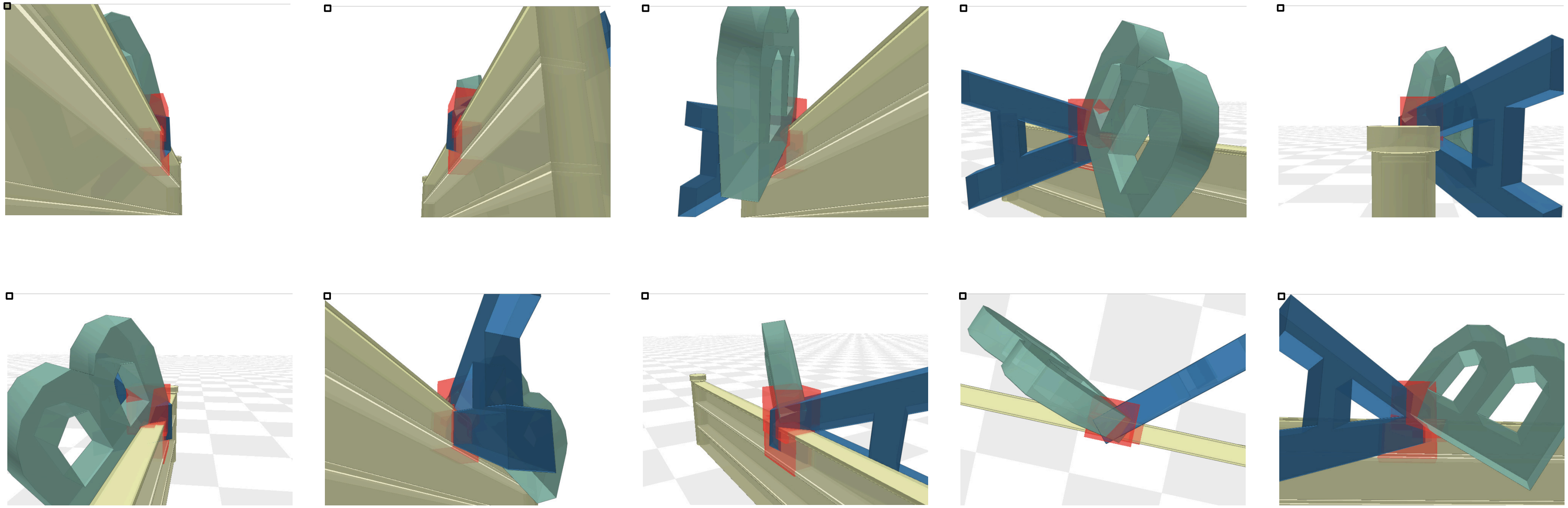


partially occluded

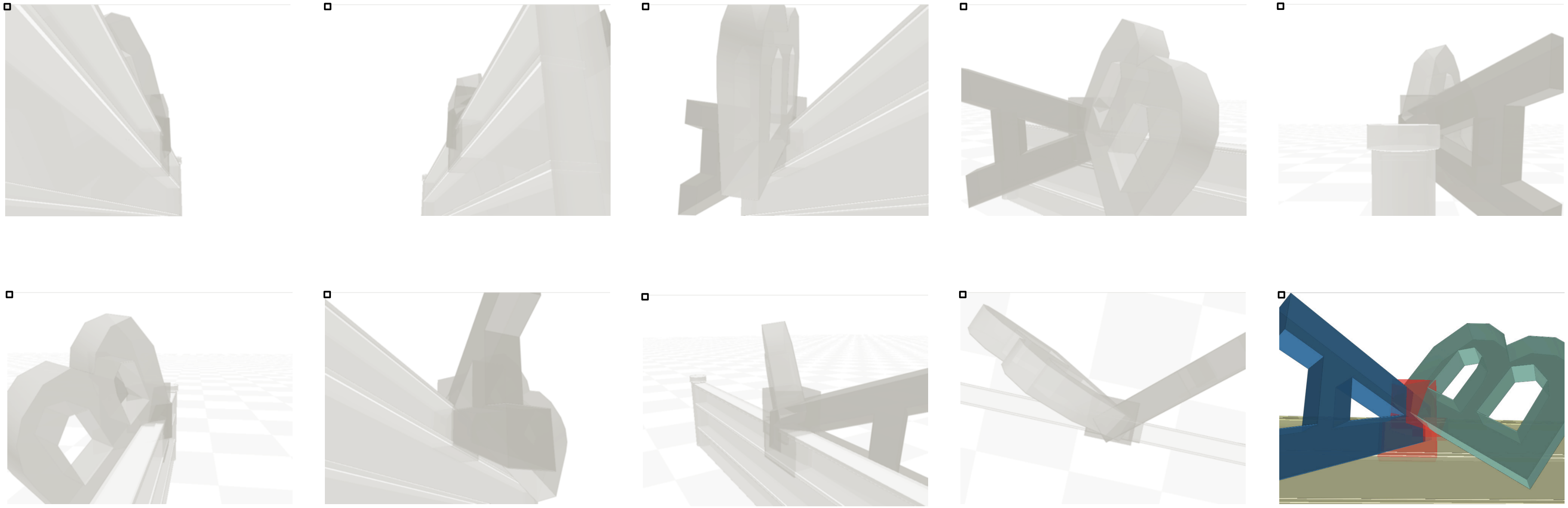
$$-\mathbf{r}_{\mathcal{A}} \times \mathbf{r}_{\mathcal{B}}$$

not occluded

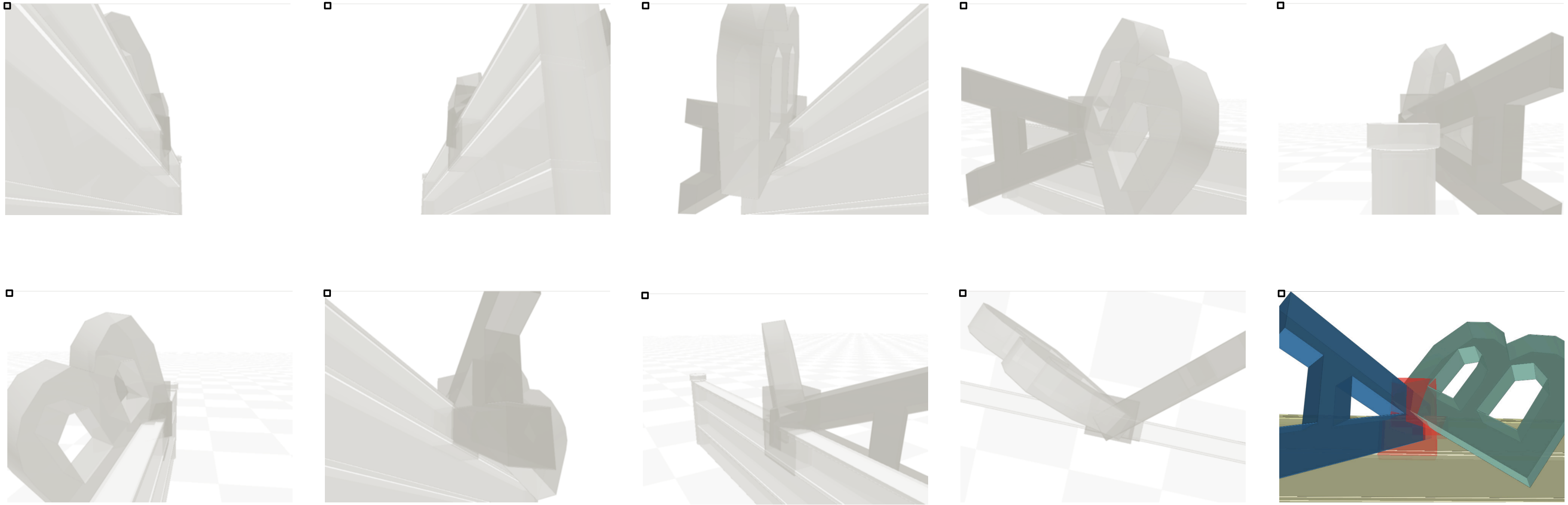
For degenerate views or multiple objects, we search over all views via *Monte Carlo*



For degenerate views or multiple objects, we search over all views via *Monte Carlo*

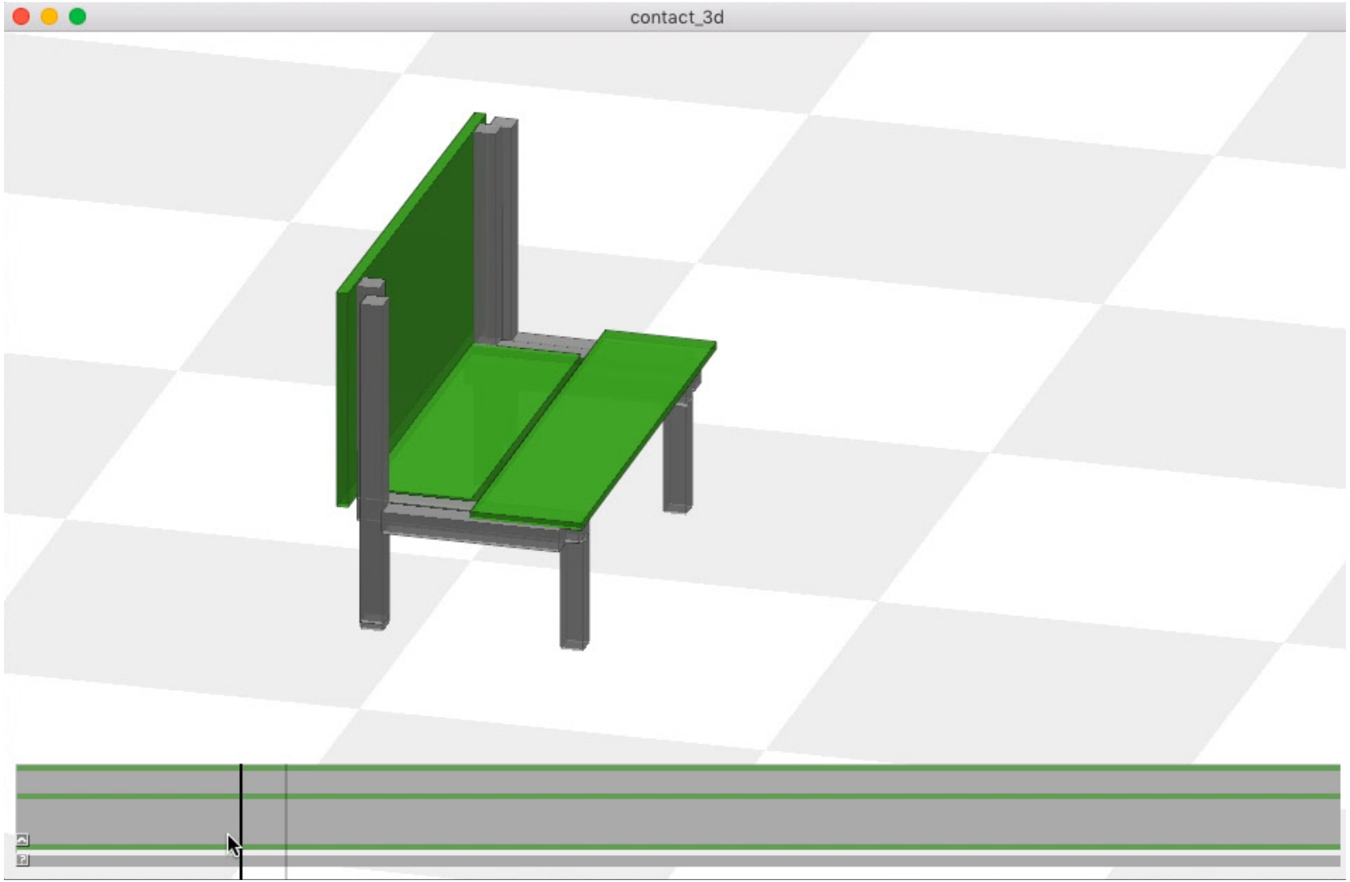


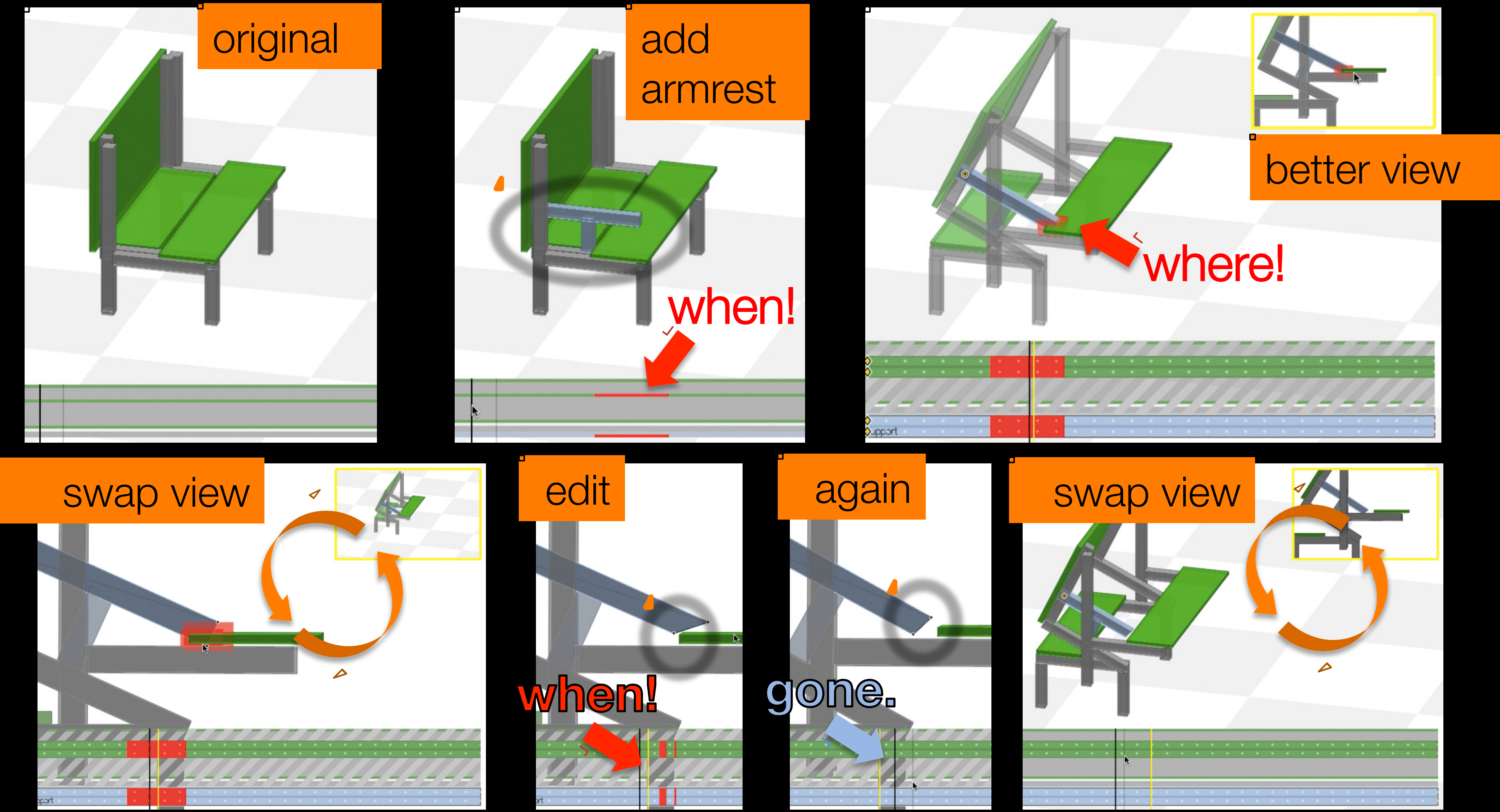
For degenerate views or multiple objects, we search over all views via *Monte Carlo*

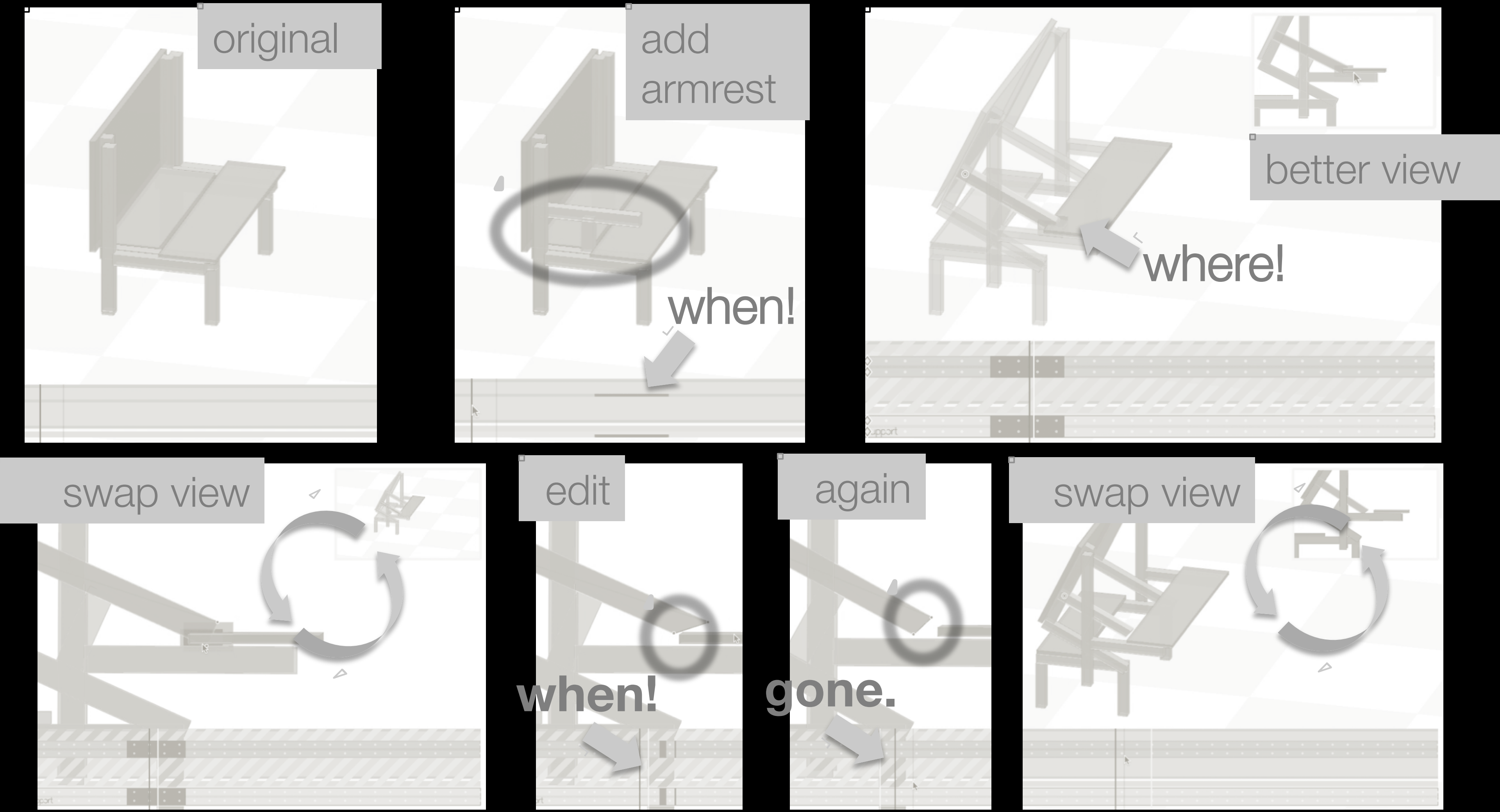


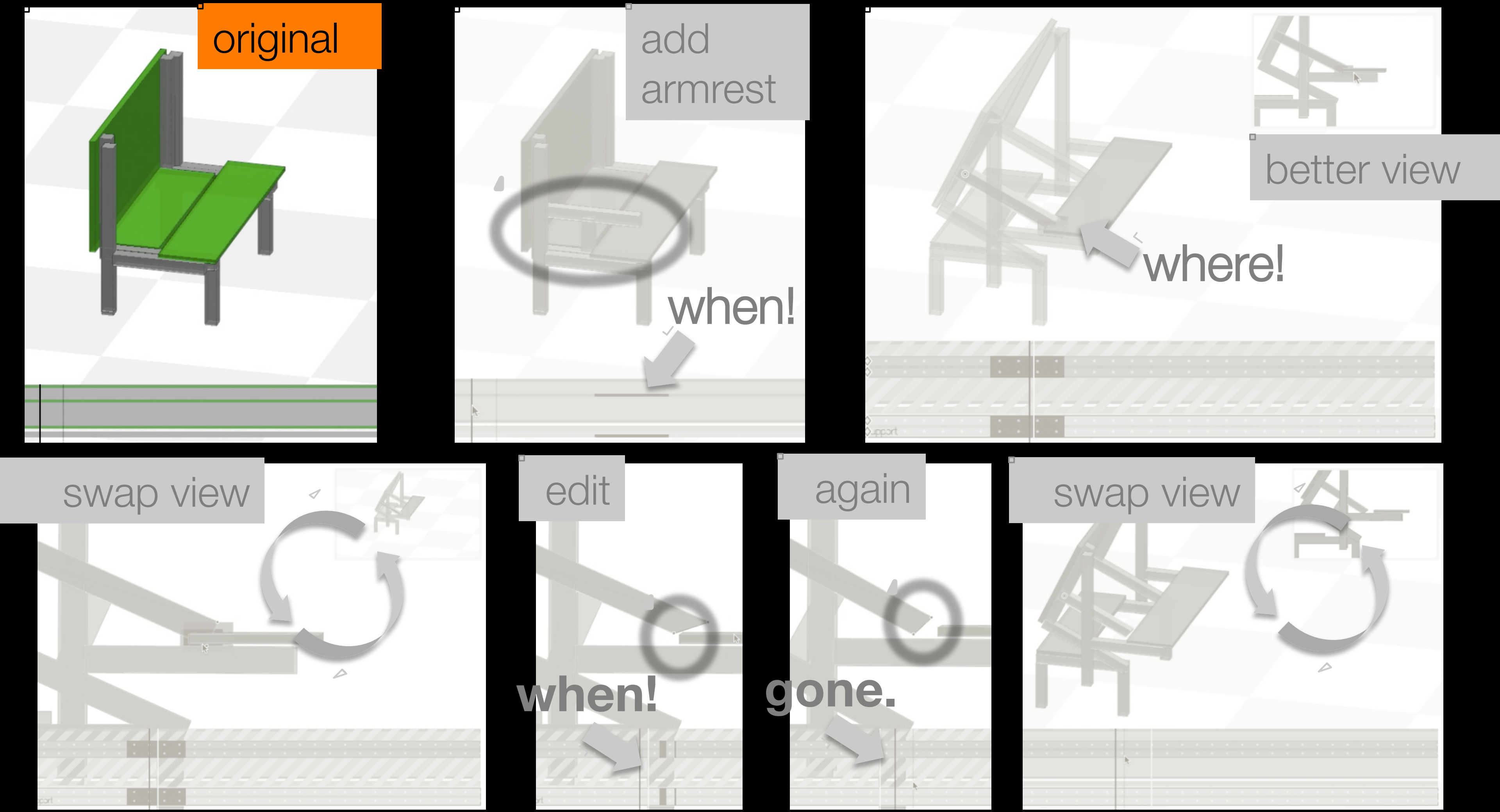
count “red” pixels using hardware “occlusion queries” (GL_SAMPLES_PASSED)

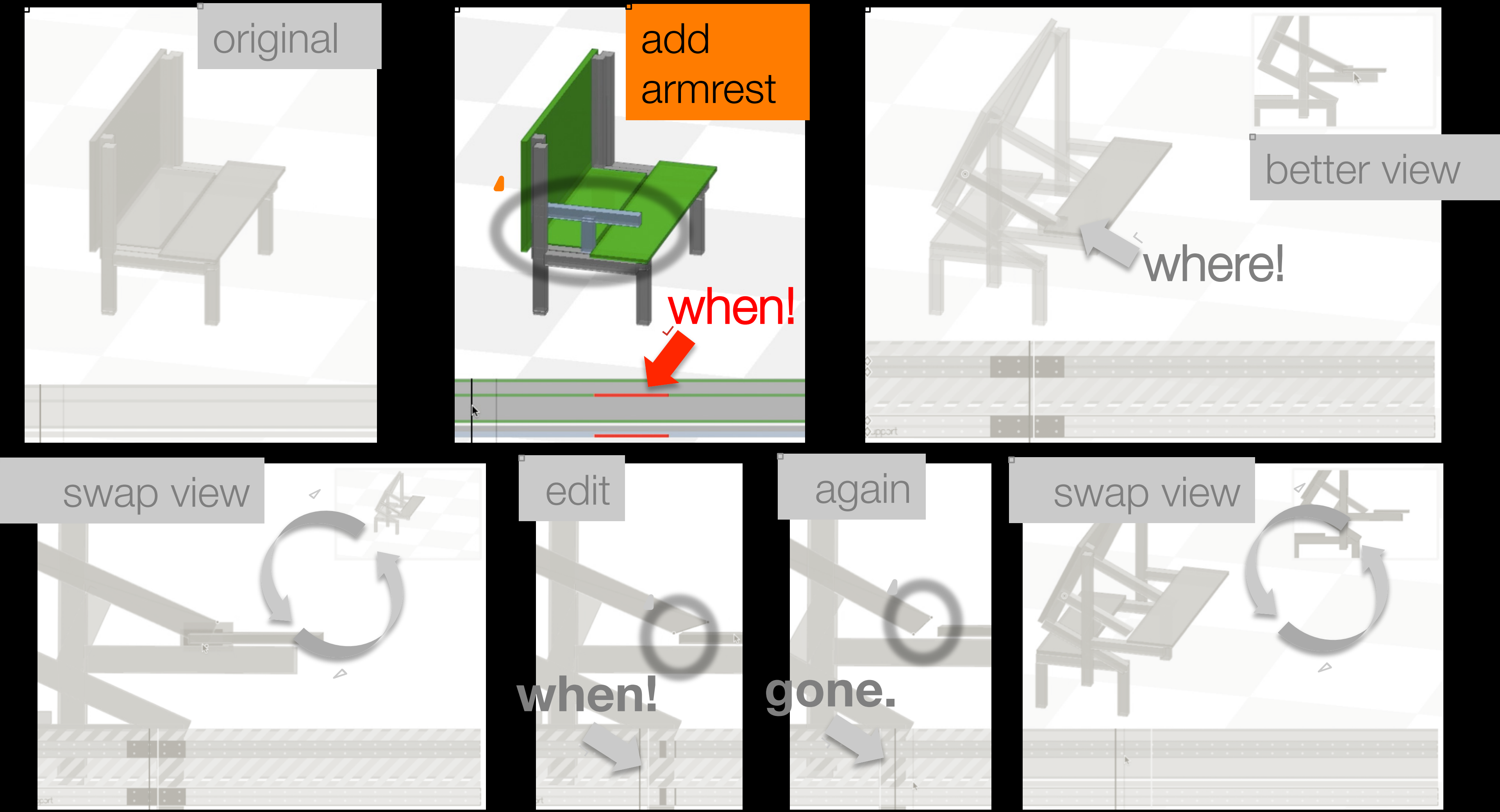
Picture-in-picture allows rapid editing

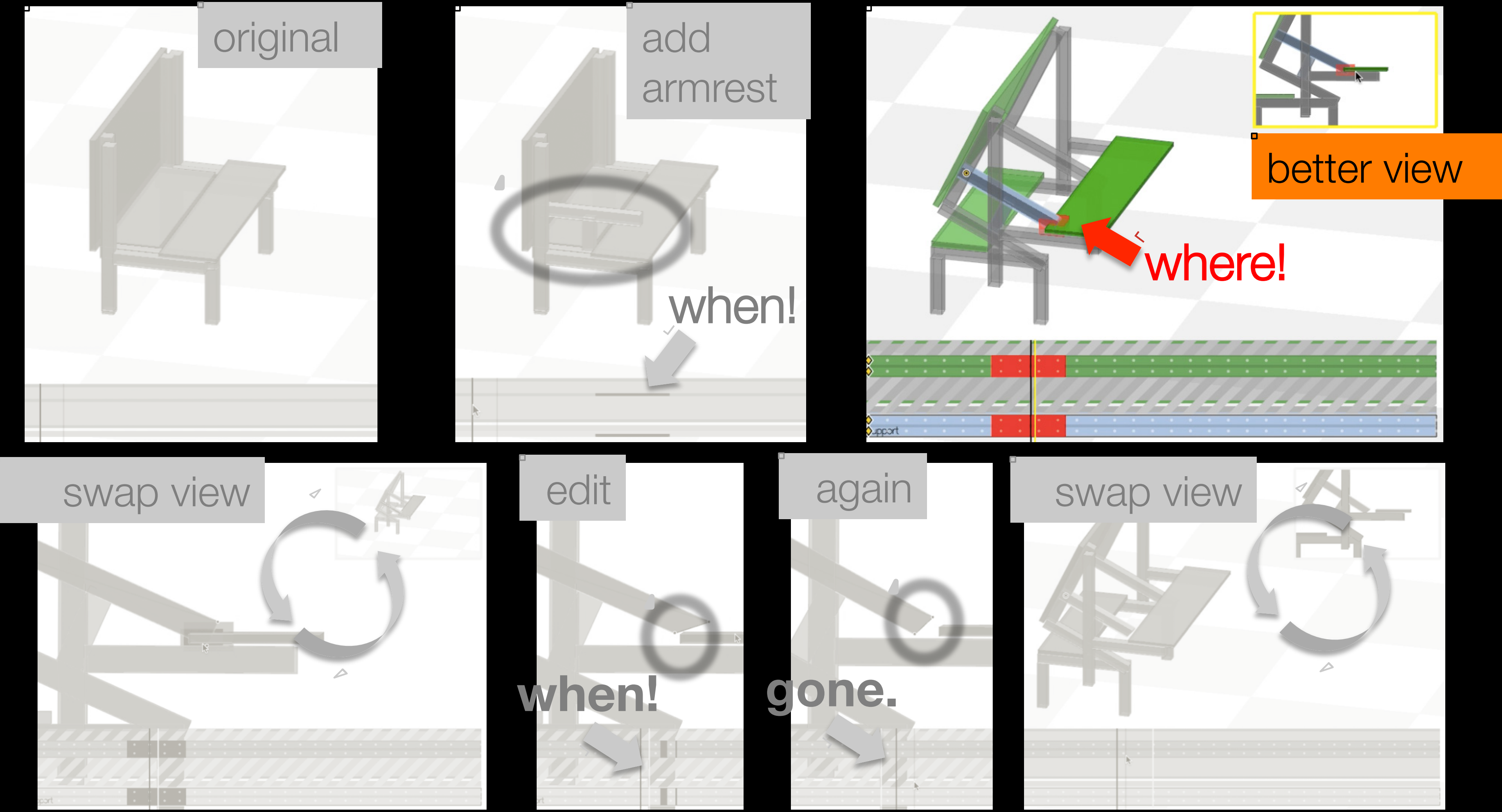


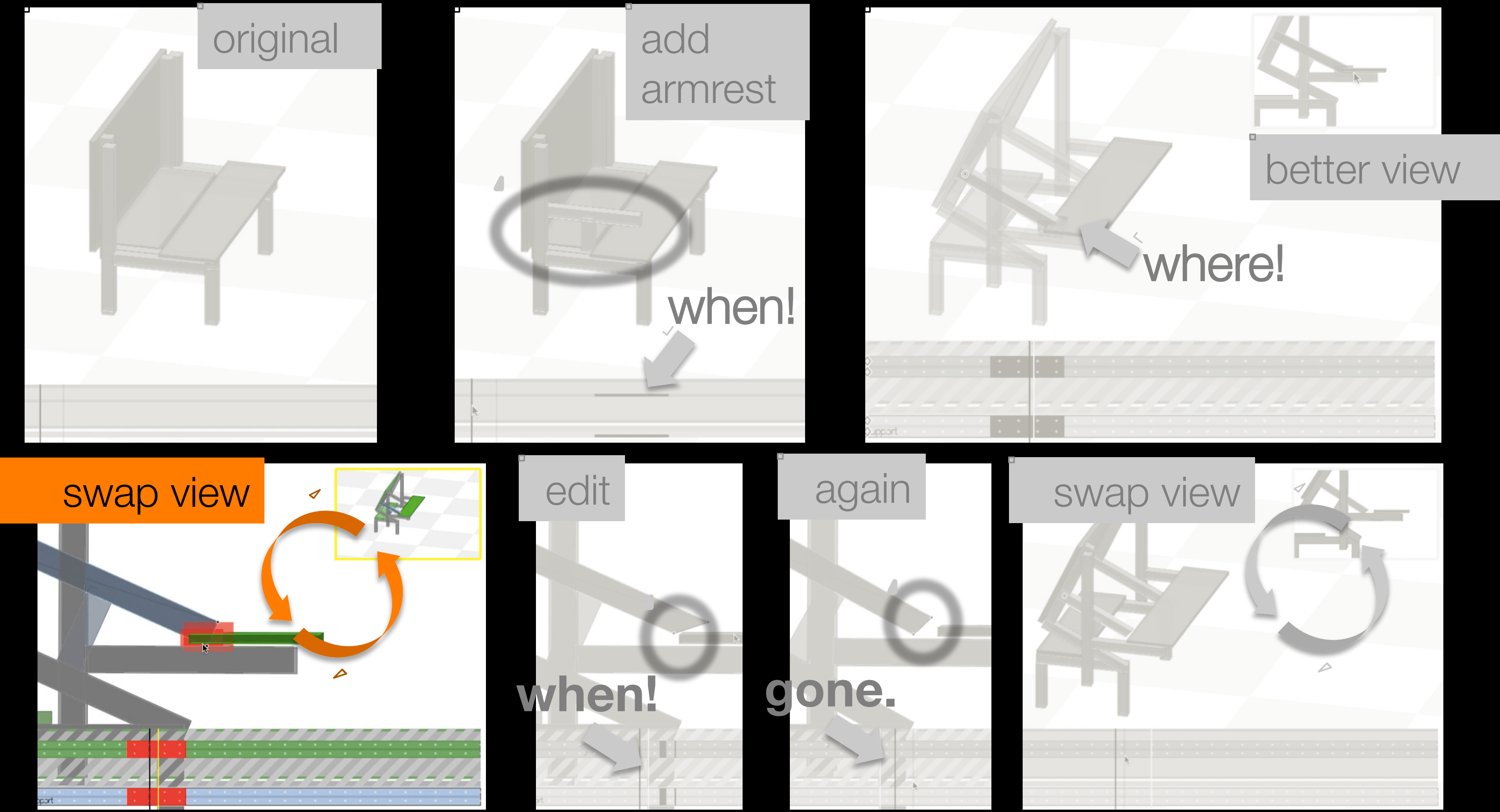

original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

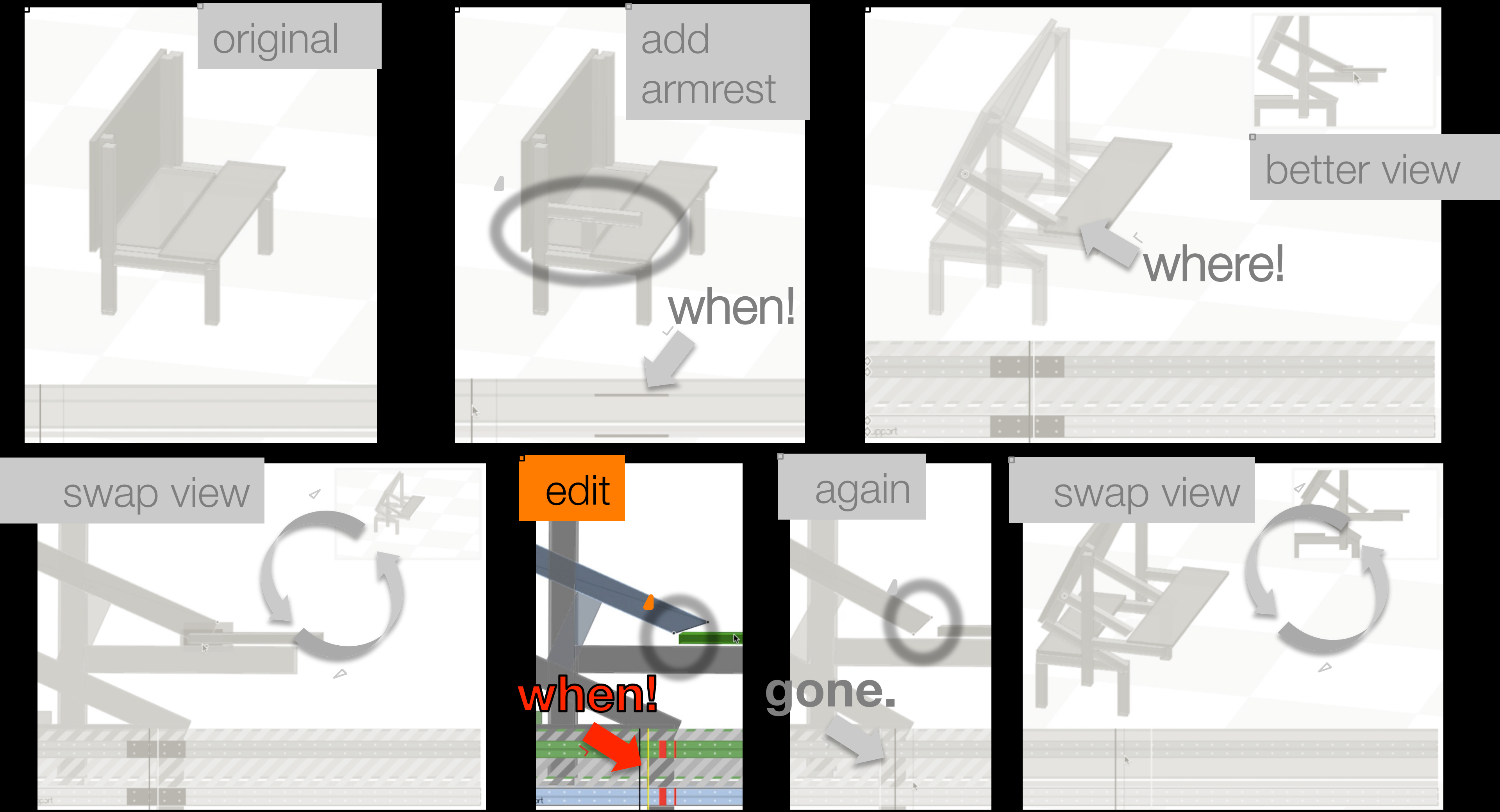

original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view


original
add
armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

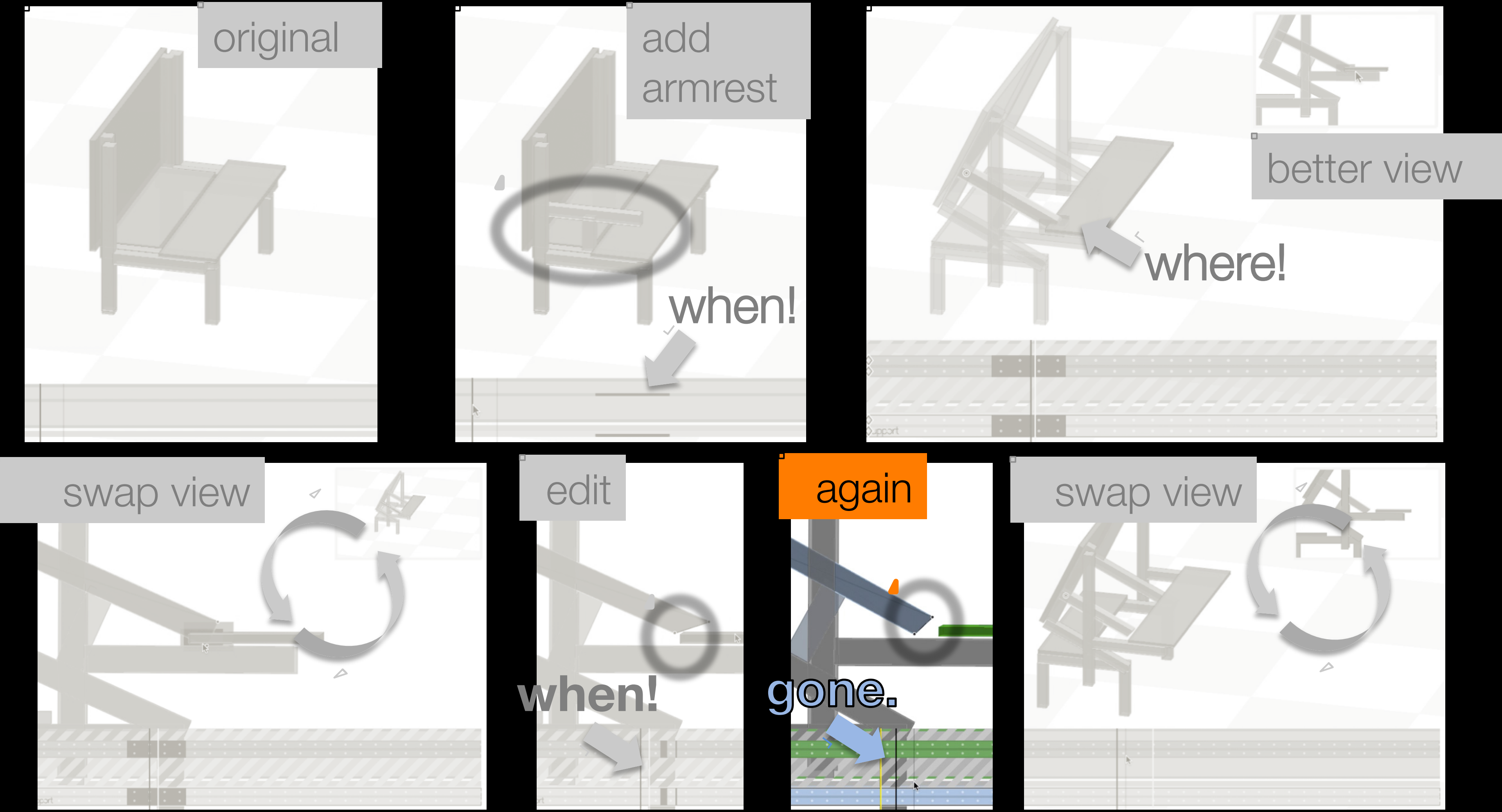

original
add
armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

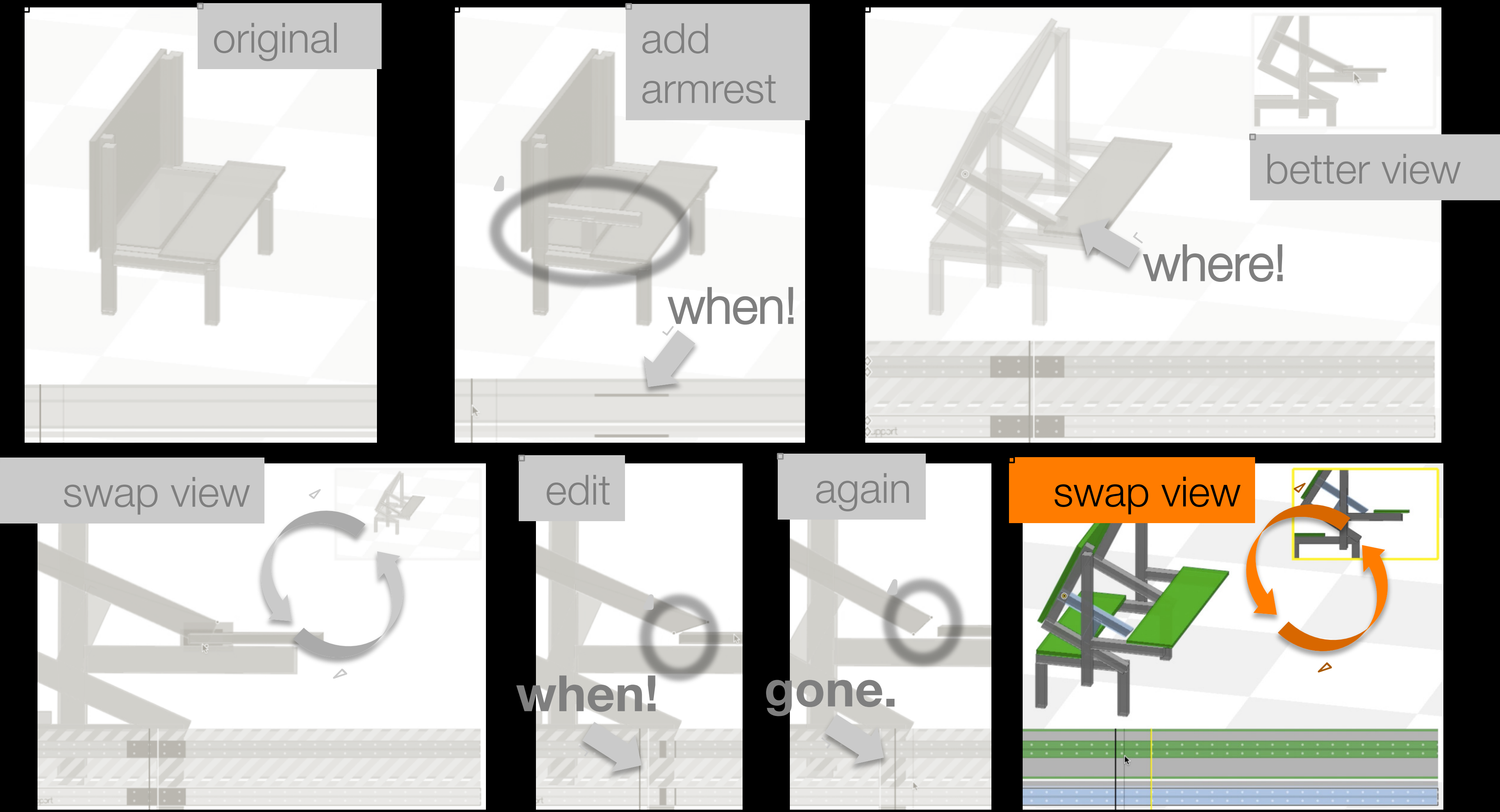

original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view


original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

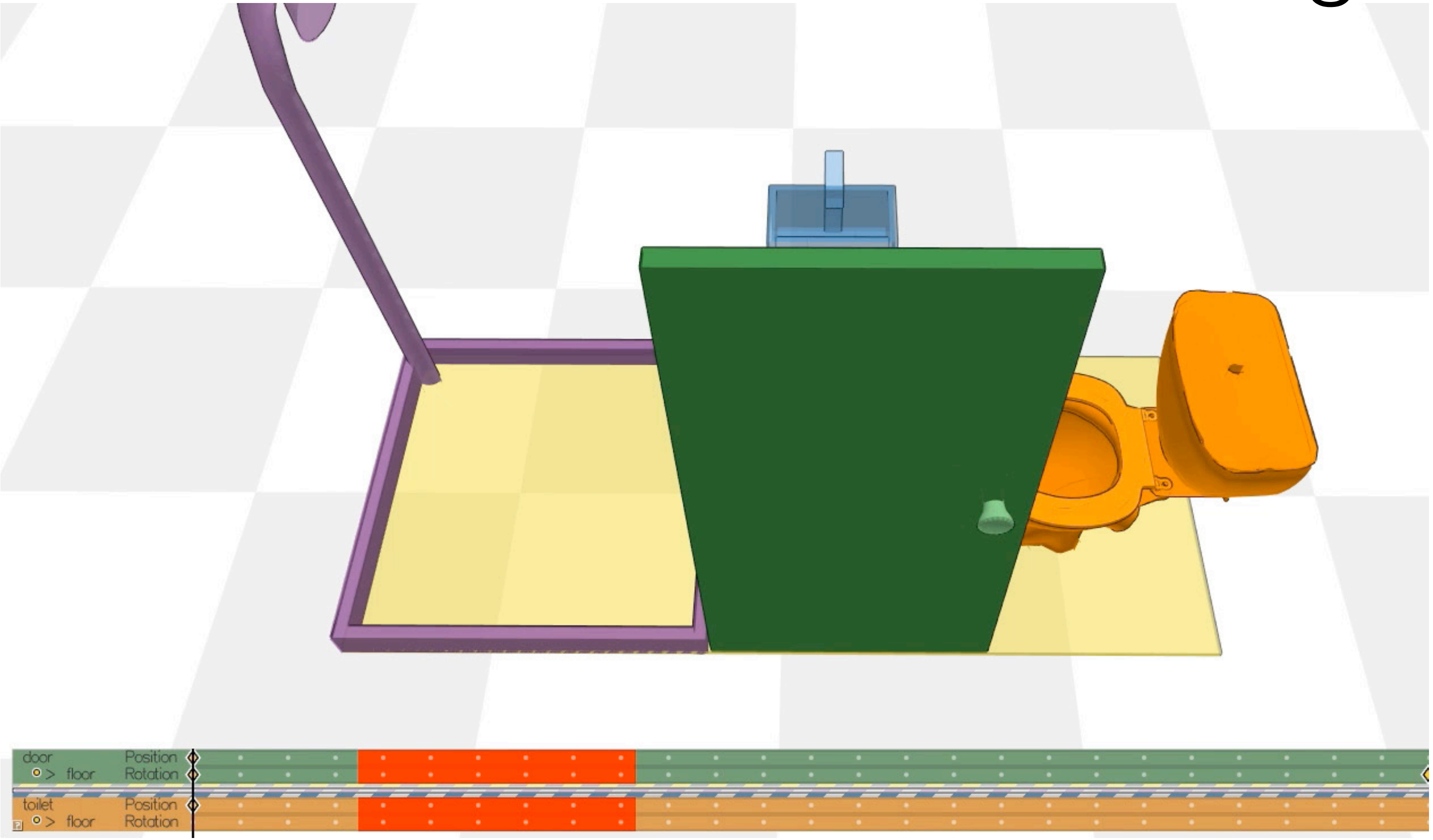


original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

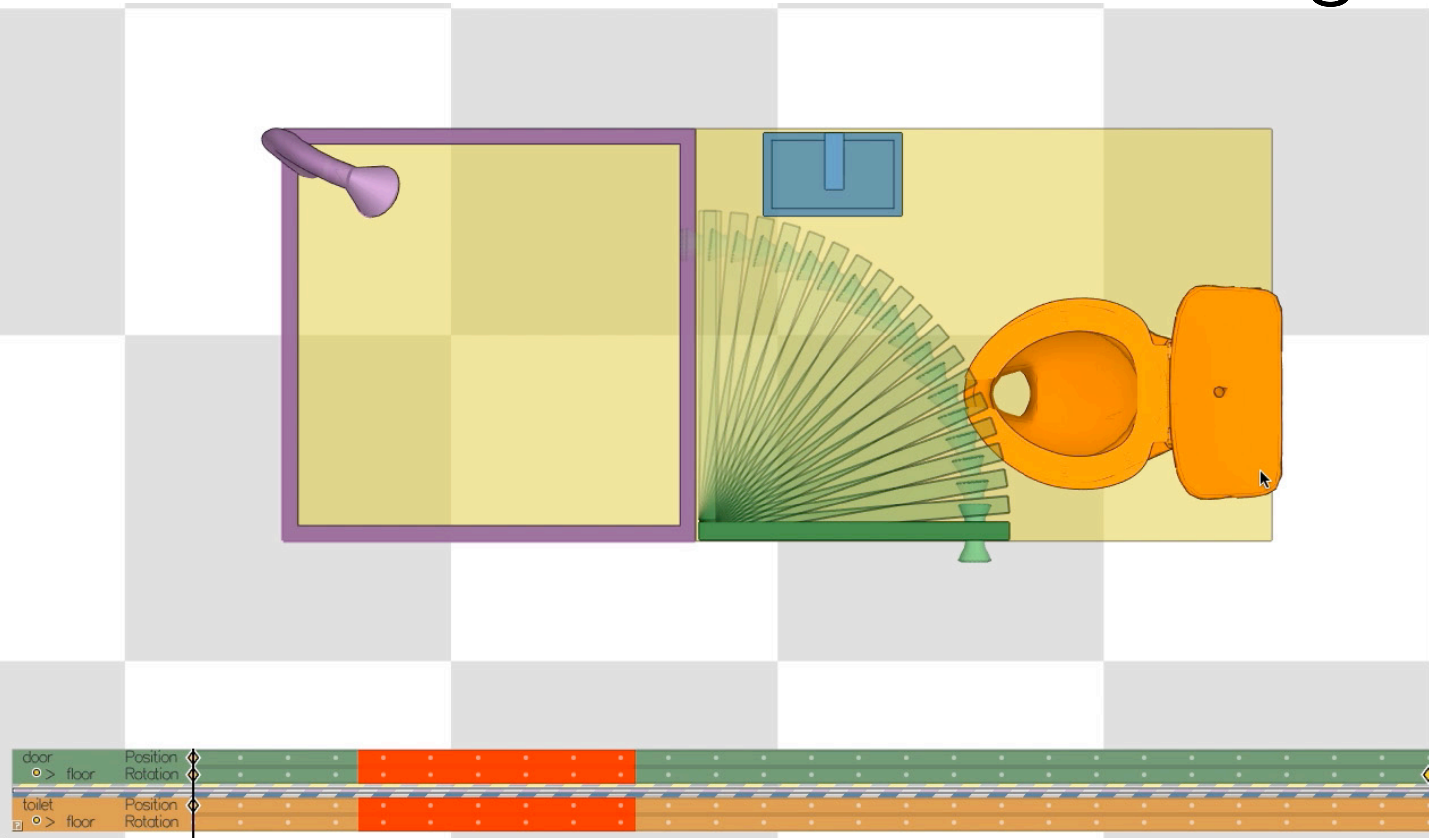

original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view


original
add armrest
when!
better view
where!
swap view
edit
when!
again
gone.
swap view

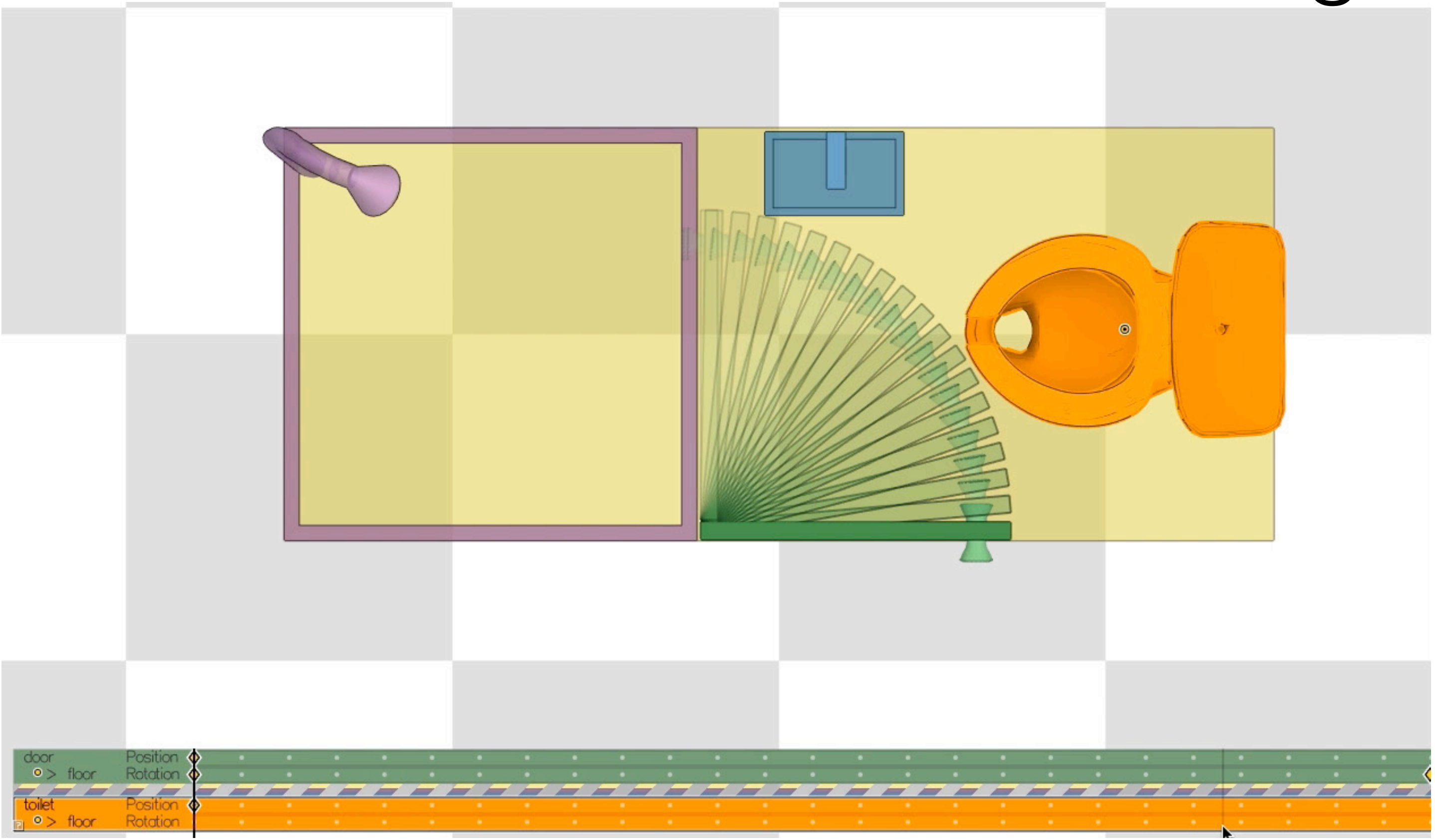
Stroboscopic ghosting complements interactive collision interval tracking



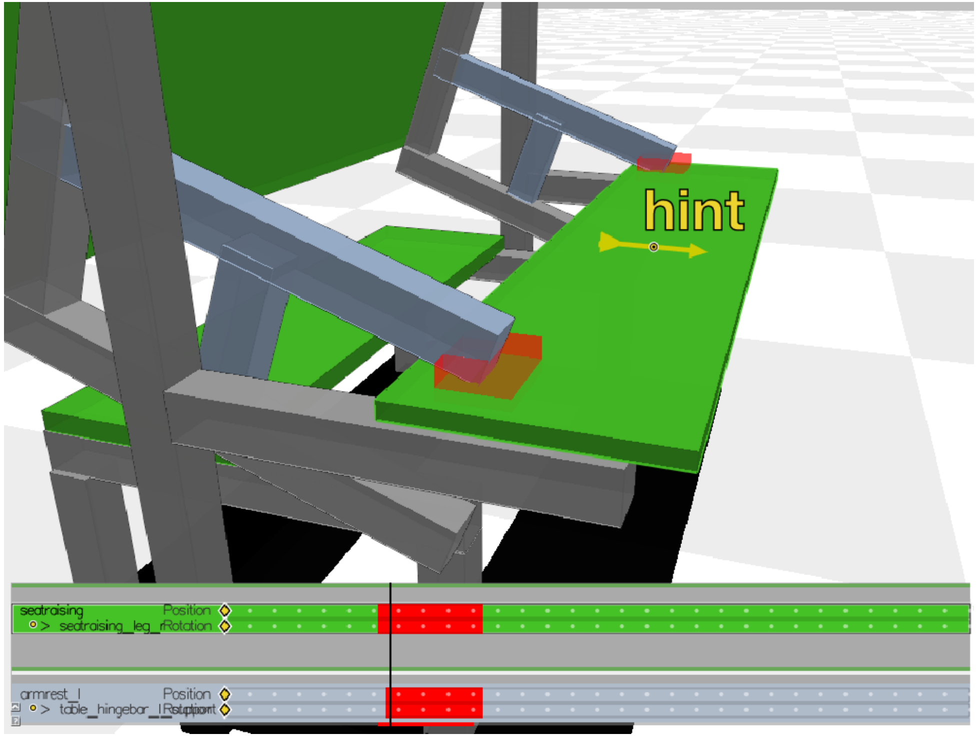
Stroboscopic ghosting complements interactive collision interval tracking



Stroboscopic ghosting complements interactive collision interval tracking



Fixing collisions can be tedious and unintuitive, how about a hint?

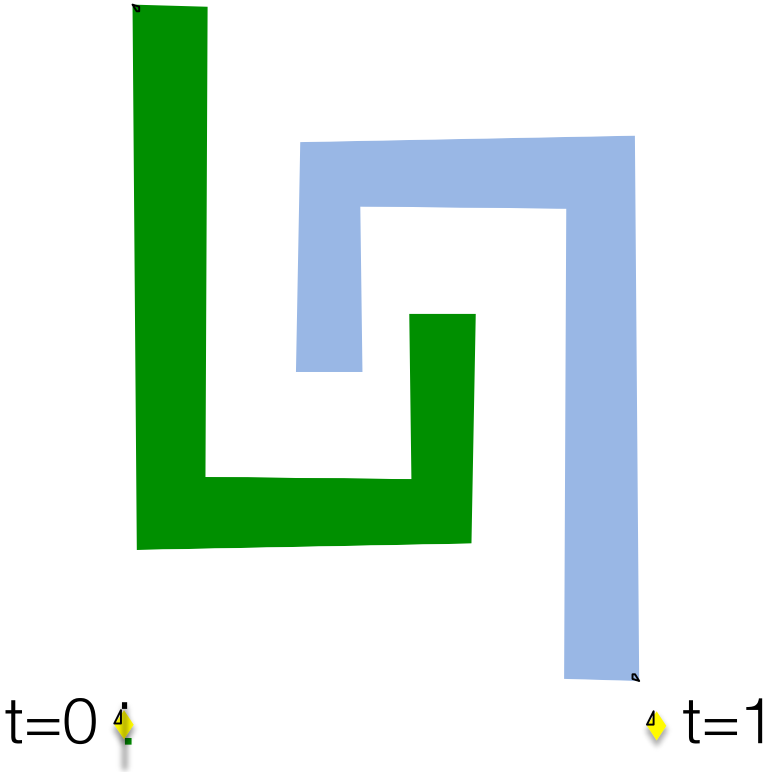


decrease/remove collisions

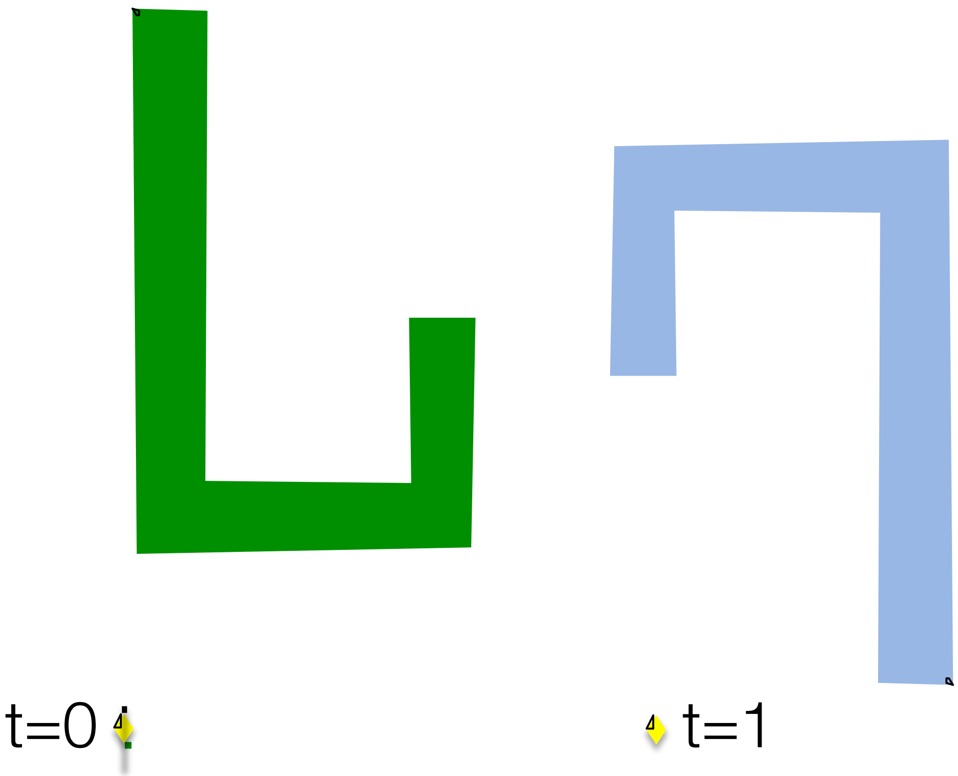
agnostic to direction of time

in terms of UI DOFs

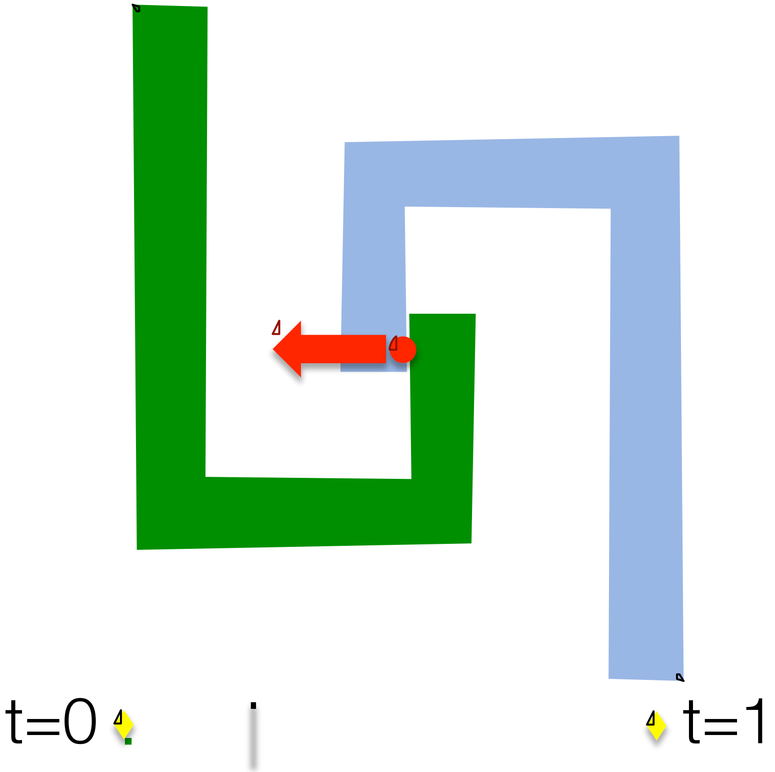
First contacts between objects produce unreliable, biased directions



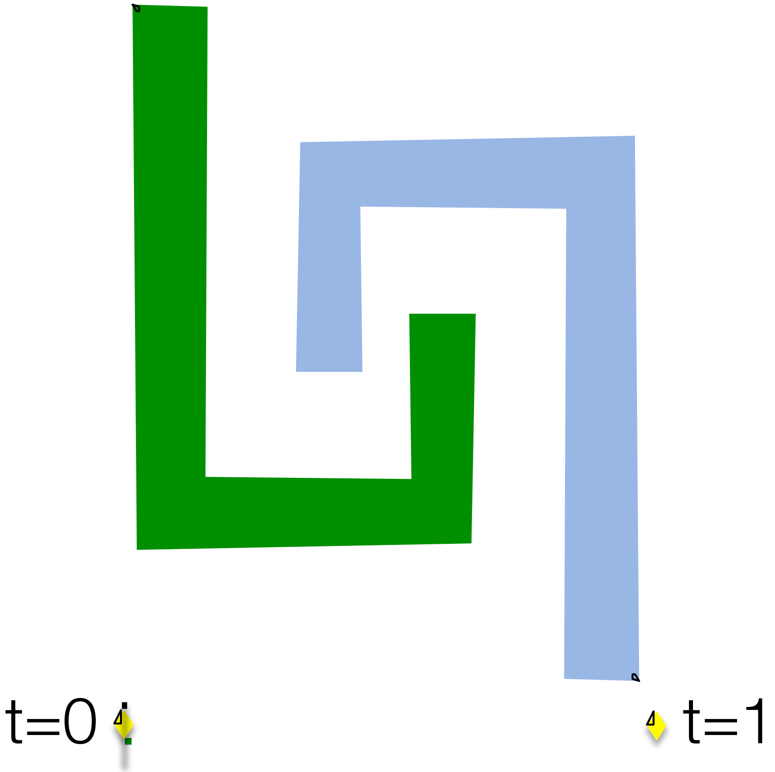
First contacts between objects produce unreliable, biased directions



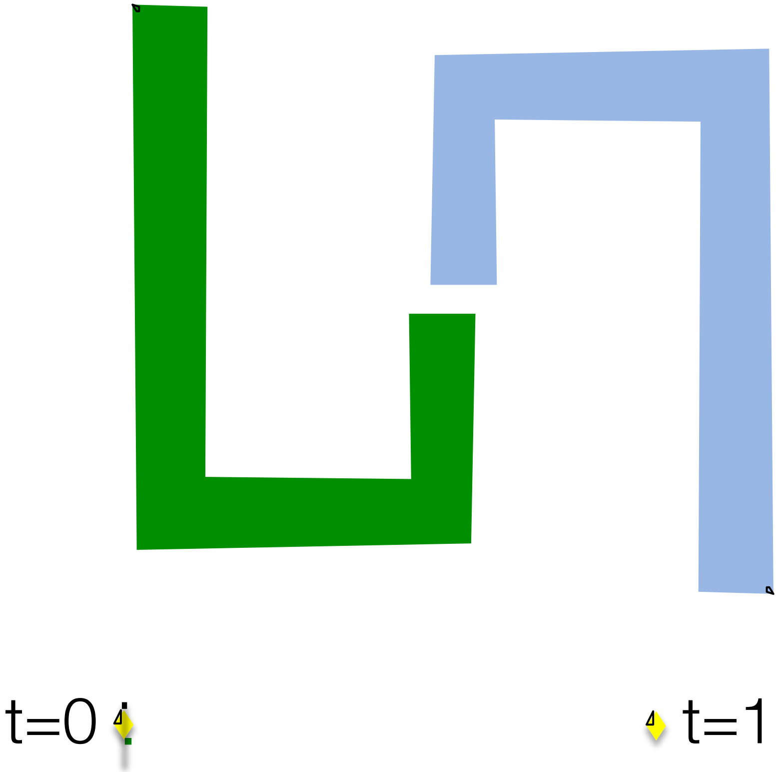
First contacts between objects produce unreliable, biased directions



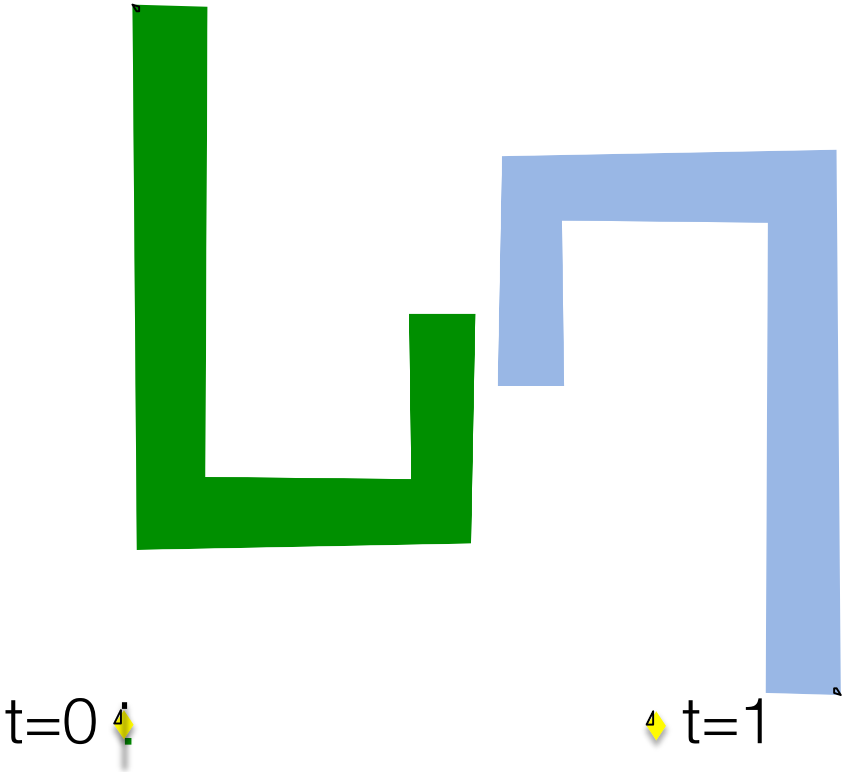
Global solution should *untangle* spacetime intersection



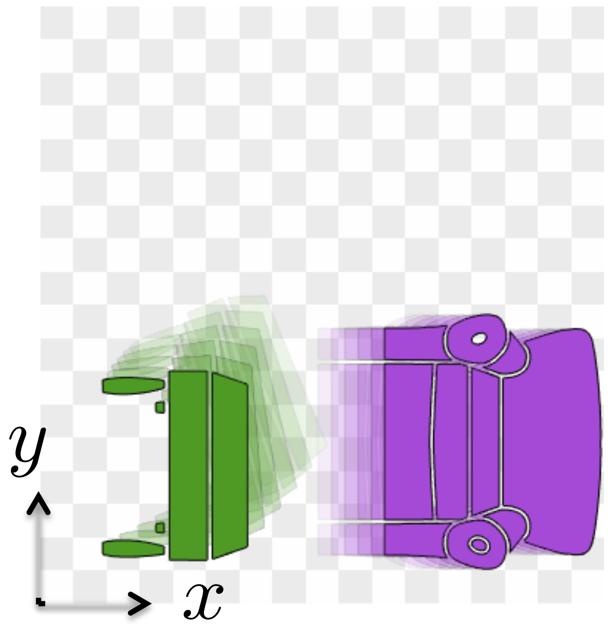
Global solution should *untangle* spacetime intersection



Global solution should *untangle* spacetime intersection

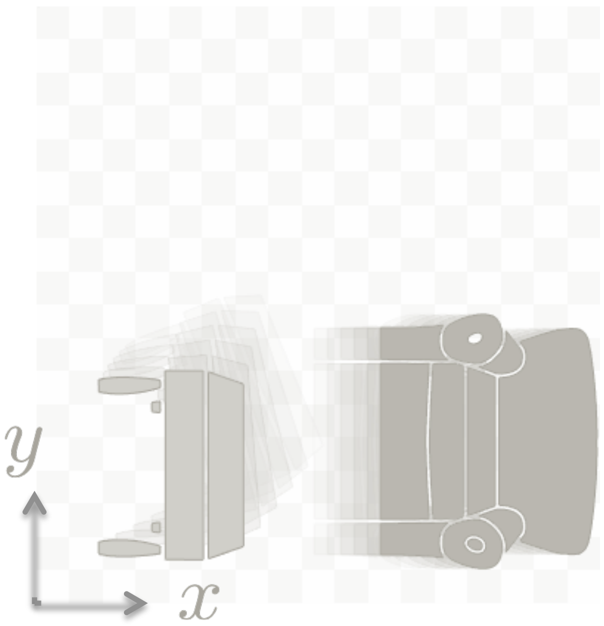


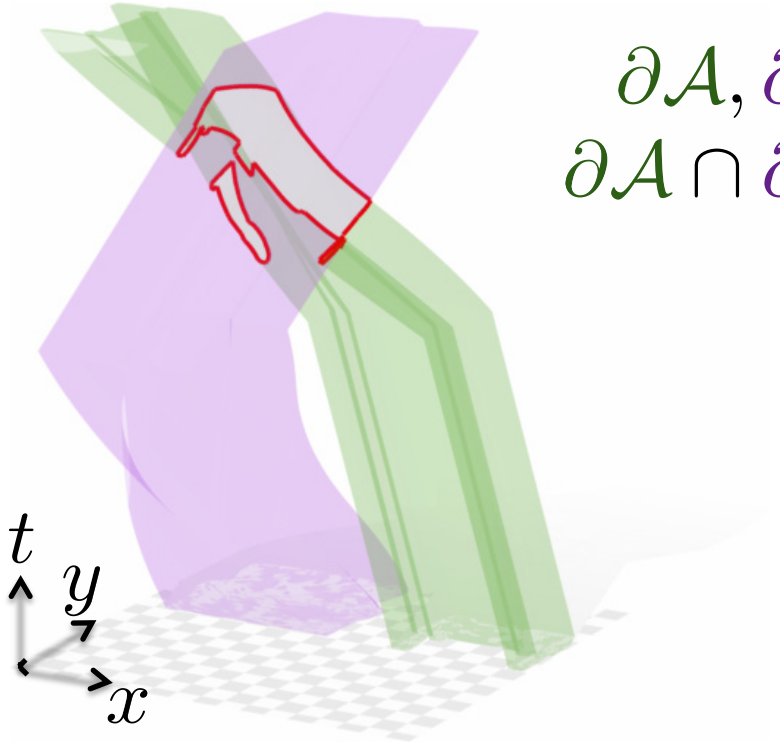
2D reconfigurables *could* be untangled like 3D cloths



$\mathcal{A}, \mathcal{B} \subset \mathbb{R}^3$

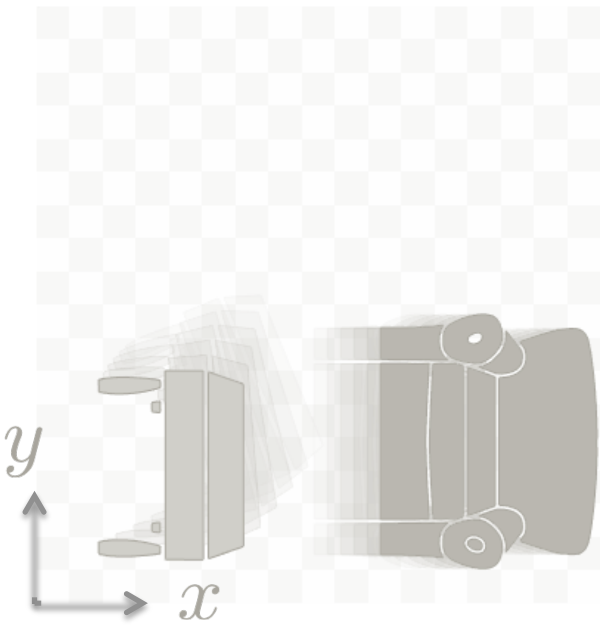
2D reconfigurables *could* be untangled like 3D cloths

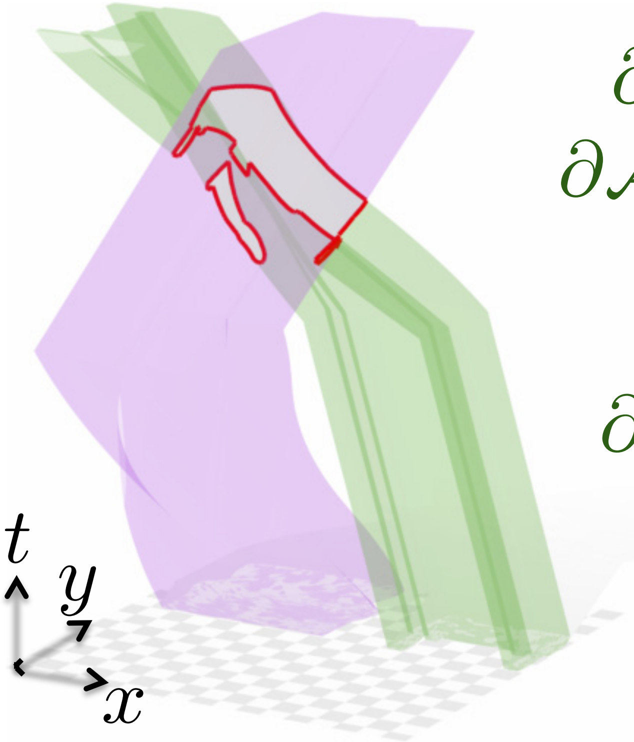




$$\partial\mathcal{A}, \partial\mathcal{B} \in \mathcal{S}_{2\mathrm{D}}$$
$$\partial\mathcal{A} \cap \partial\mathcal{B} \in \mathcal{C}_{1\mathrm{D}}$$

2D reconfigurables *could* be untangled like 3D cloths

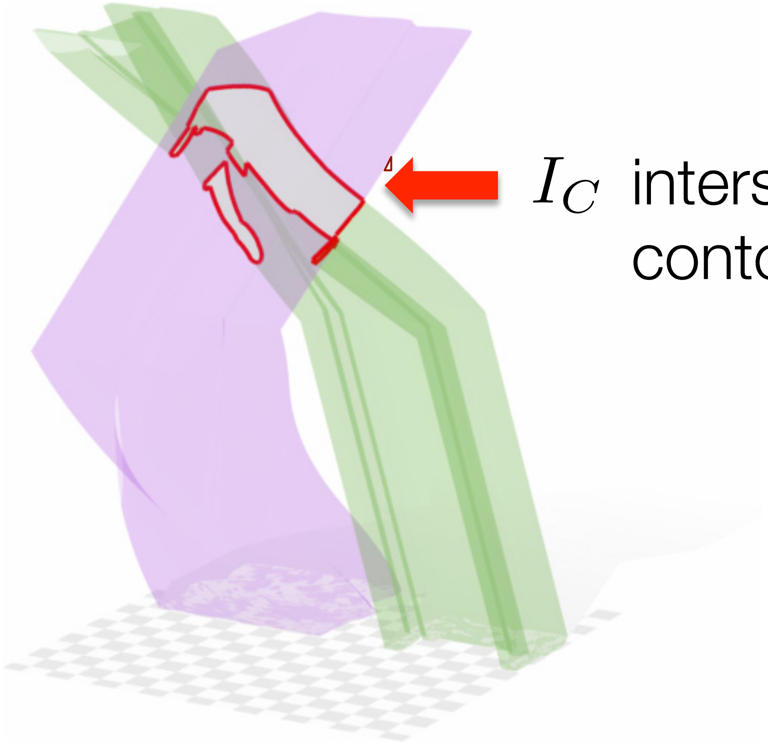




$$\partial\mathcal{A}, \partial\mathcal{B} \in \mathcal{S}_{\mathrm{2D}}$$
$$\partial\mathcal{A} \cap \partial\mathcal{B} \in \mathcal{C}_{\mathrm{1D}}$$

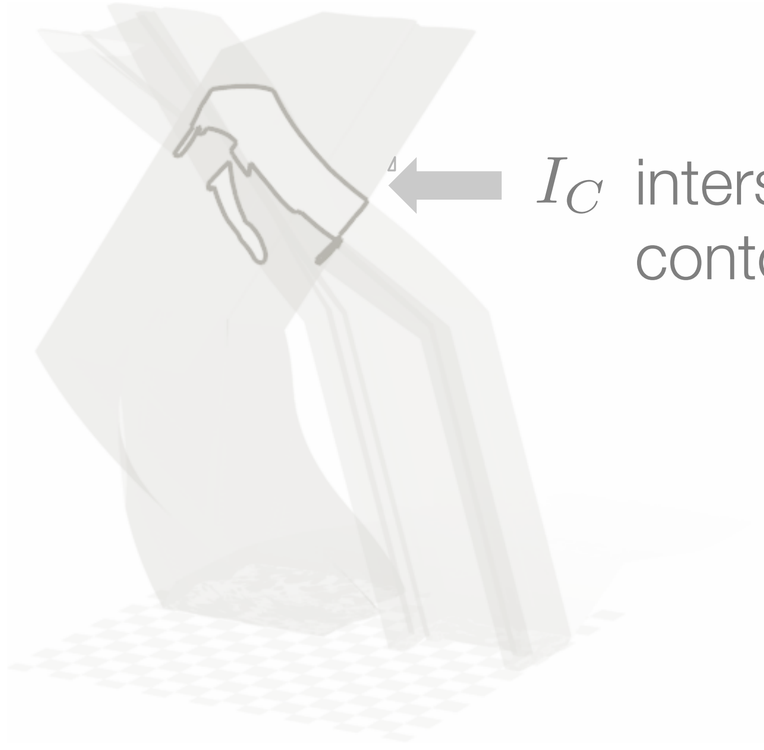
$$\partial\mathcal{A} \cap \partial\mathcal{B} \to \varnothing$$
$$\Downarrow$$
$$\mathcal{A} \cap \mathcal{B} \to \varnothing$$

Minimizing length of intersection curve results in untangling



I_C intersection contour

Minimizing length of intersection curve results in untangling



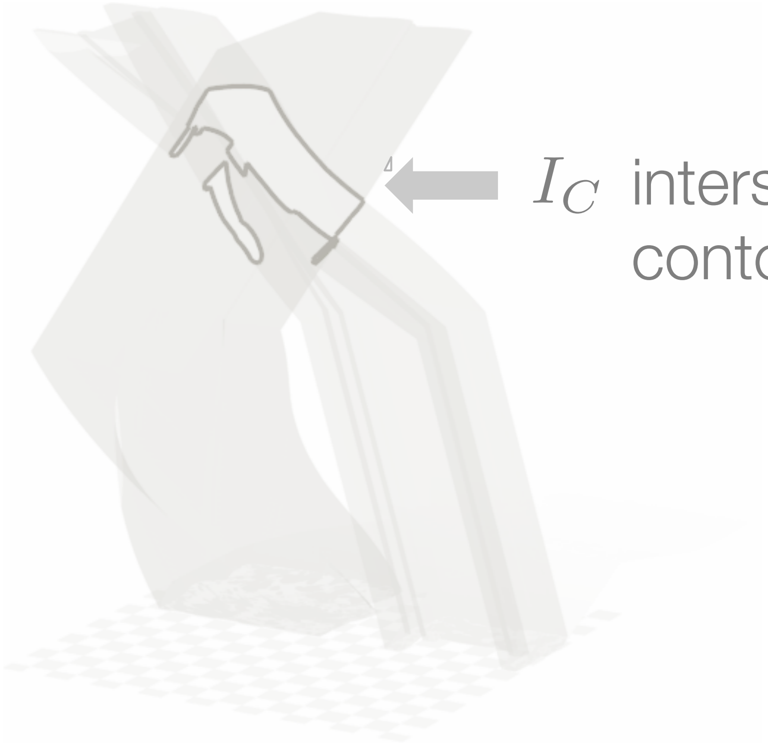
I_C intersection contour

$$\underset{\mathbf{V}}{\operatorname{argmin}} \; \|I_C\|$$

3D mesh vertices

length

Minimizing length of intersection curve results in untangling



I_C intersection contour

$$\underset{\mathbf{V}}{\operatorname{argmin}} \; \|I_C\|$$

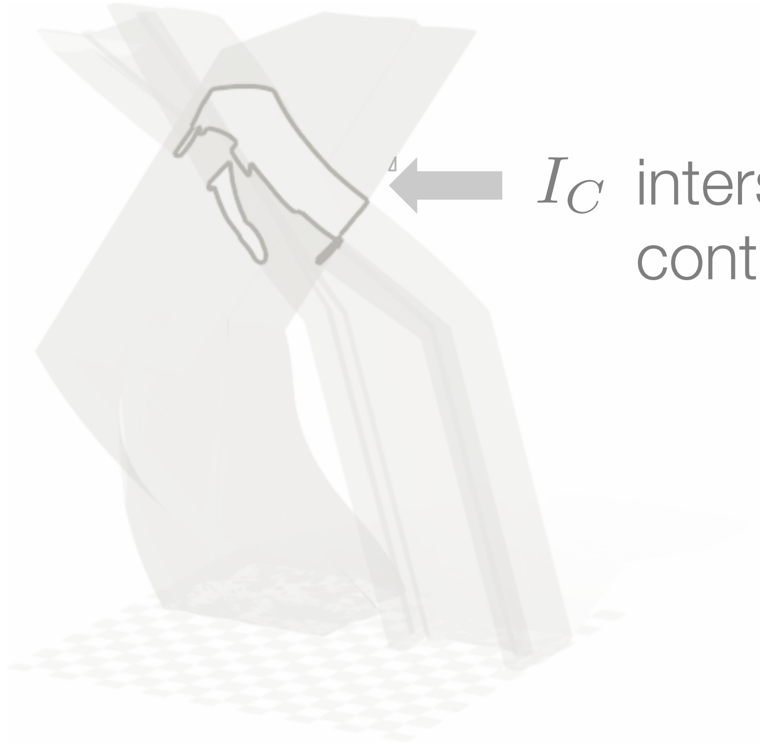
3D mesh vertices

length

update via gradient descent

$$\mathbf{V} \leftarrow \mathbf{V} - \nabla_{\mathbf{V}} \|I_C\|$$

Minimizing length of intersection curve results in untangling



I_C intersection contour

$$\underset{\mathbf{V}}{\operatorname{argmin}} \; \|I_C\|$$

3D mesh vertices

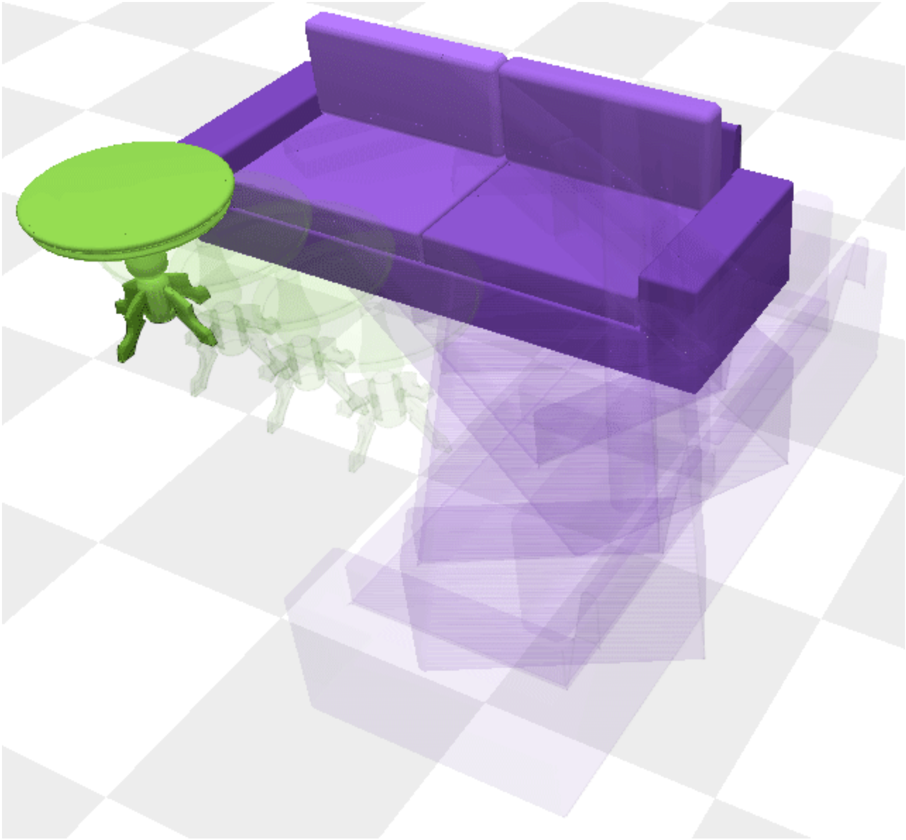
length

update via gradient descent

$$\mathbf{V} \leftarrow \mathbf{V} - \boxed{\nabla_{\mathbf{V}} \|I_C\|}$$

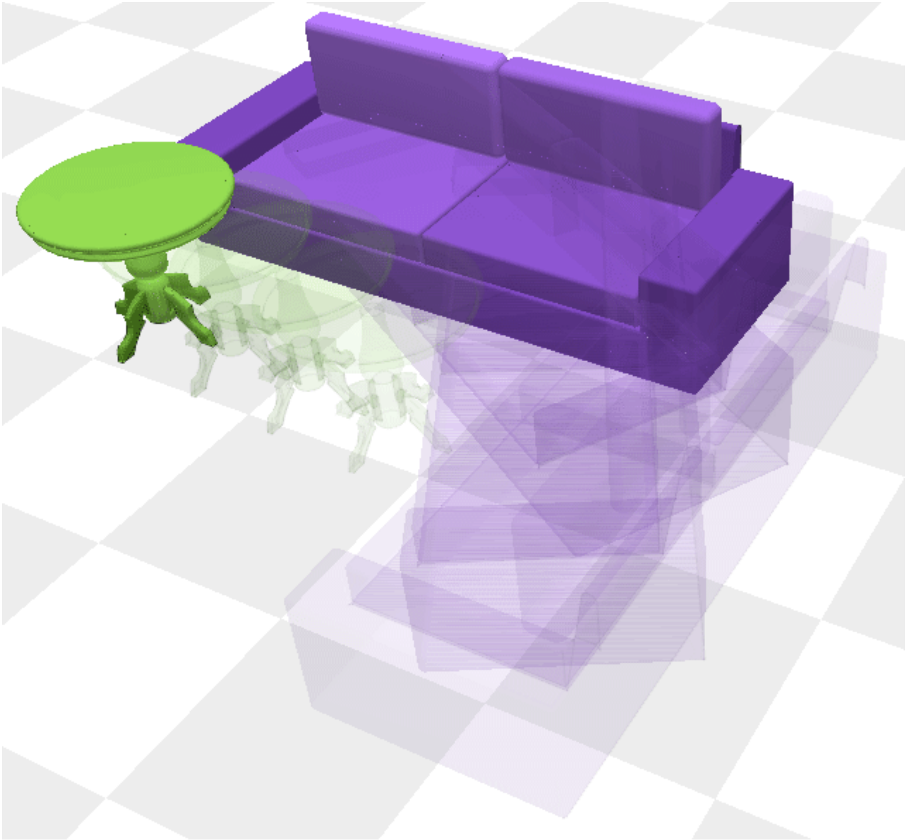
[Volino & M-T 2006]

Naive extension to 3D objects is problematic



$$\mathcal{A}, \mathcal{B} \subset \mathbb{R}^4$$

Naive extension to 3D objects is problematic

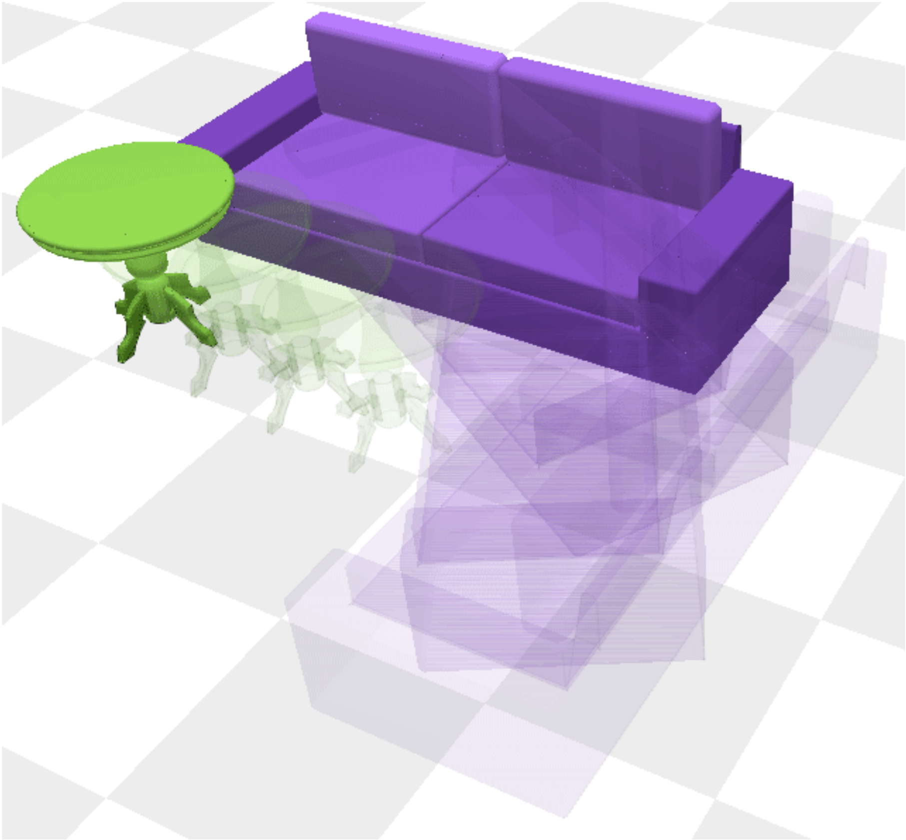


$$\partial\mathcal{A}, \partial\mathcal{B} \in \mathcal{S}_{3D}$$

$$\partial\mathcal{A} \cap \partial\mathcal{B} \in \mathcal{S}_{2D}$$

Difficult to compute or optimize over

Naive extension to 3D objects is problematic



$$\partial\mathcal{A}, \partial\mathcal{B} \in \mathcal{S}_{3\text{D}}$$

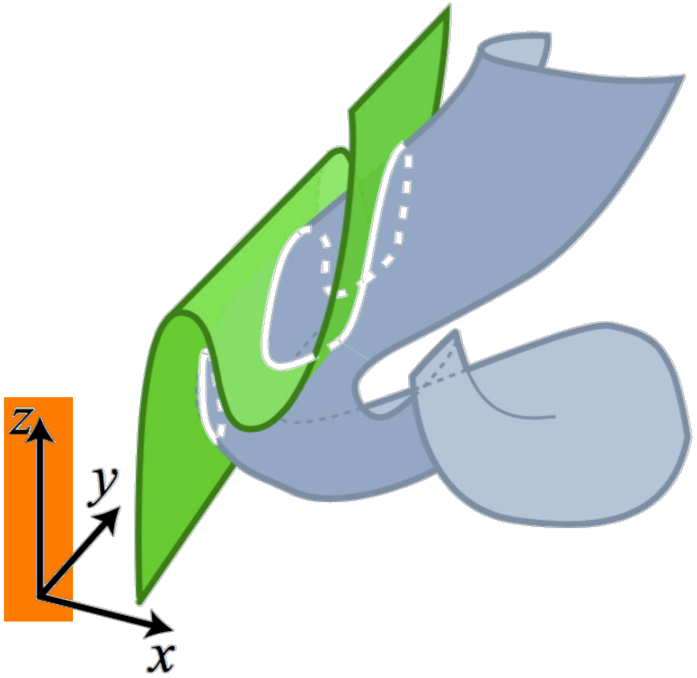
$$\partial\mathcal{A} \cap \partial\mathcal{B} \in \mathcal{S}_{2\text{D}},$$

Difficult to compute or optimize over

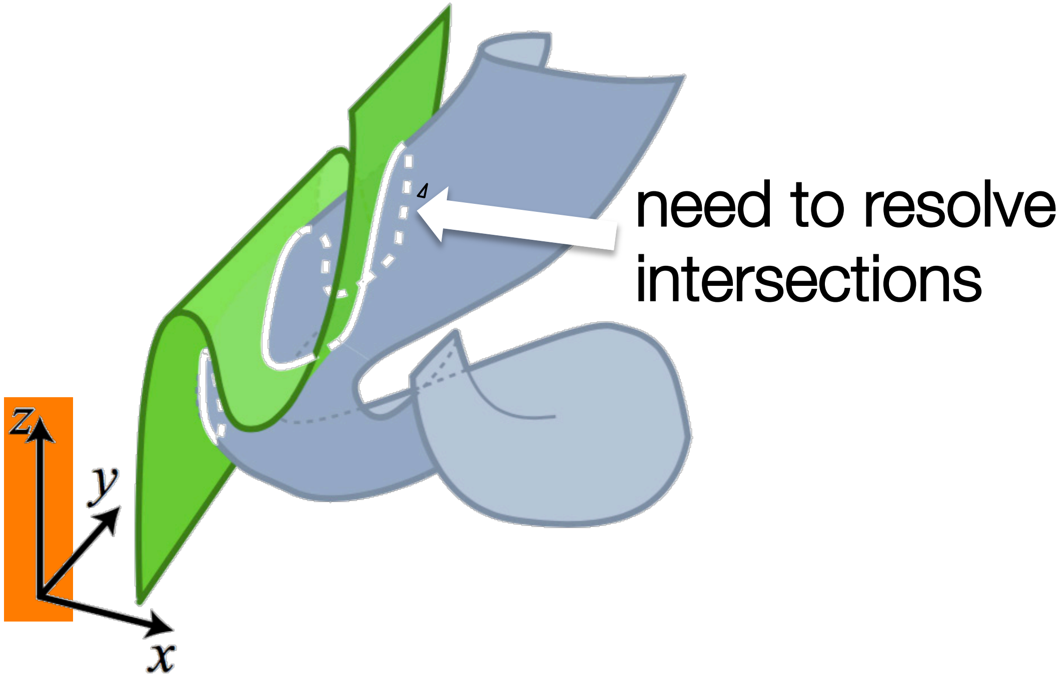
How to minimize surface area of intersection?

$$\underset{\text{UI}}{\operatorname{argmin}} \; |I_S|$$

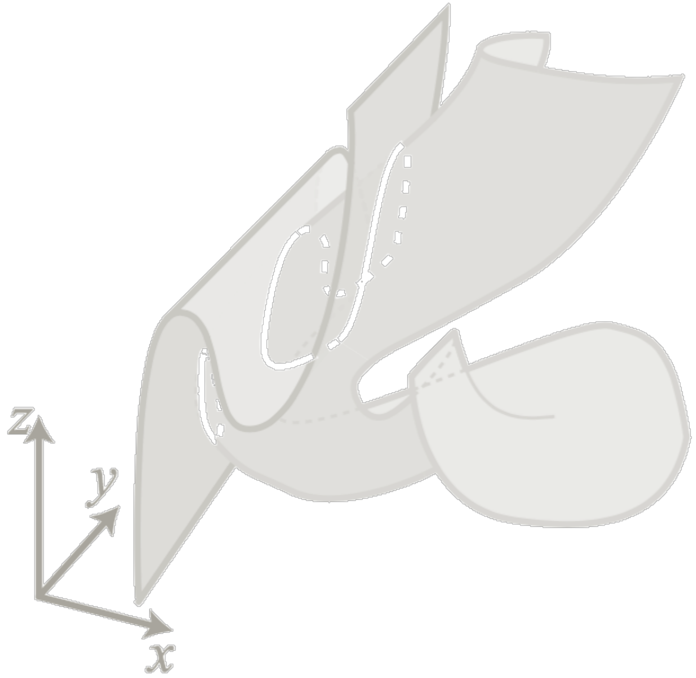
Intersections of generic surfaces are non-trivial to parameterize…

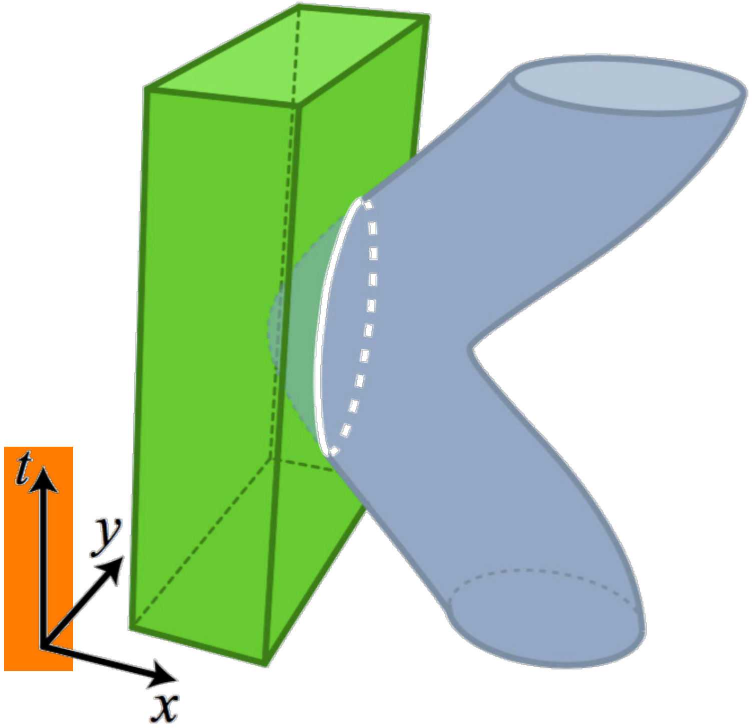


Intersections of generic surfaces are non-trivial to parameterize…

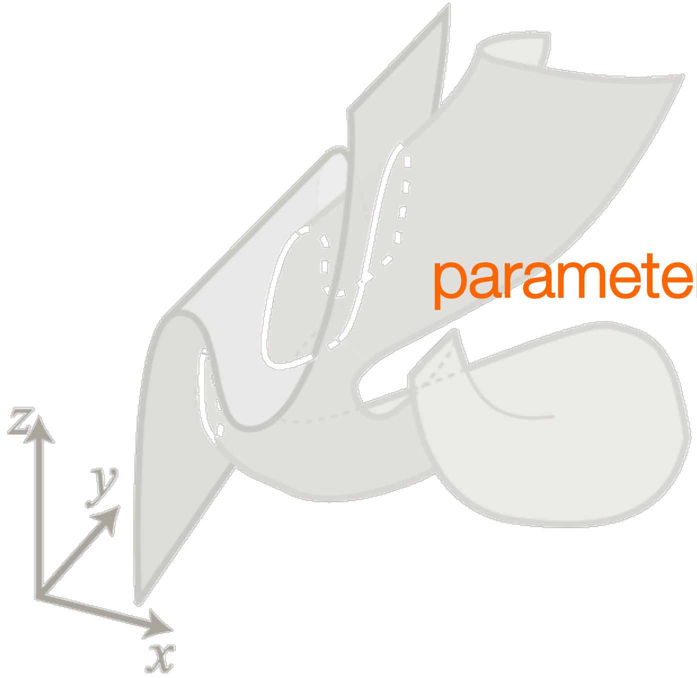


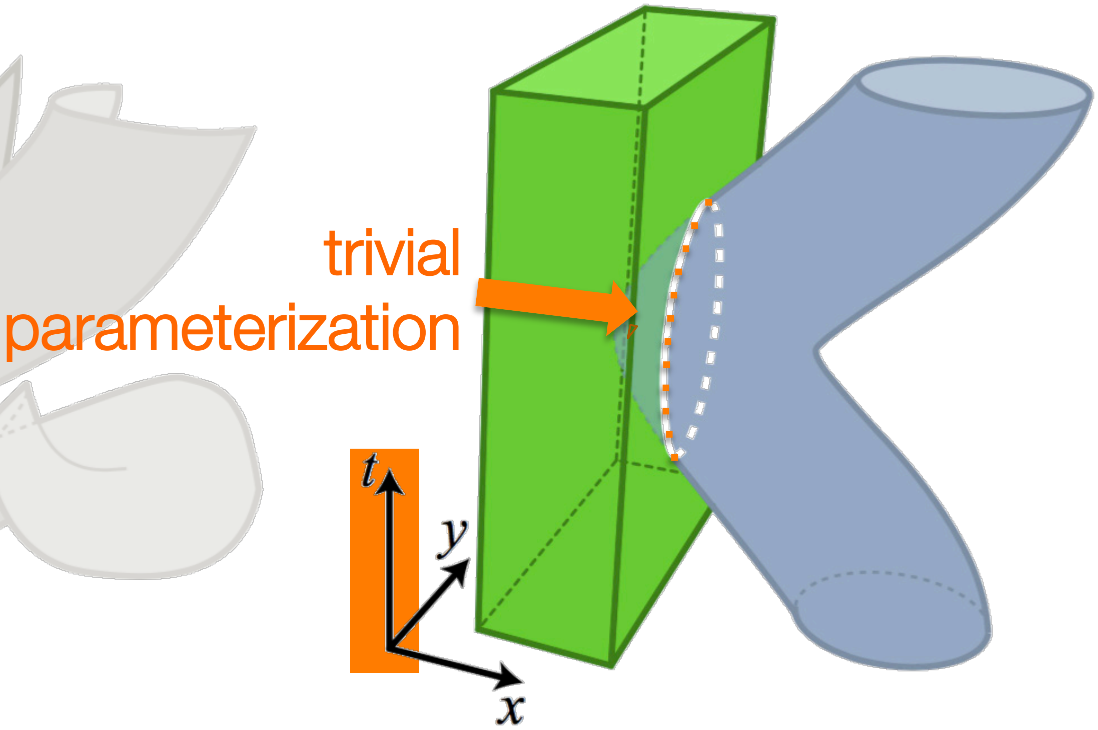
… but spacetime surfaces are explicit functions of monotonic time





… but spacetime surfaces are explicit functions of monotonic time





Spacetime intersection surface area reduces to integral of 1D intersection contours

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \; |I_S|$$

$$\mathrm{UI} \leftarrow \mathrm{UI} - \nabla_{\mathrm{UI}} |I_S|$$

Spacetime intersection surface area reduces to integral of 1D intersection contours

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \int_0^1 \|I_C(t)\| \, dt$$

$$\mathrm{UI} \leftarrow \mathrm{UI} - \int_0^1 \nabla_{\mathrm{UI}} \|I_C(t)\| \, dt$$

Applying the chain rule reveals UI couples surface positions across time

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \int_0^1 \|I_C(t)\| \, dt$$

$$\mathrm{UI} \leftarrow \mathrm{UI} - \int_0^1 \nabla_{\mathrm{UI}} \|I_C(t)\| \, dt$$

apply chain rule

Applying the chain rule reveals UI couples surface positions across time

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \int_0^1 \|I_C(t)\| \, dt$$

$$\mathrm{UI} \leftarrow \mathrm{UI} - \int_0^1 \left(\nabla_{\mathrm{UI}} \mathbf{V}_t\right) \left(\nabla_{\mathbf{V}_t} \|I_C(t)\|\right) \, dt$$

Applying the chain rule reveals UI couples surface positions across time

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \int_0^1 \|I_C(t)\| \, dt$$

$$\mathrm{UI} \leftarrow \mathrm{UI} - \int_0^1 \left(\nabla_{\mathrm{UI}} \mathbf{V}_t\right) \boxed{\left(\nabla_{\mathbf{V}_t} \|I_C(t)\|\right)} \, dt$$

[Volino & M-T 2006]

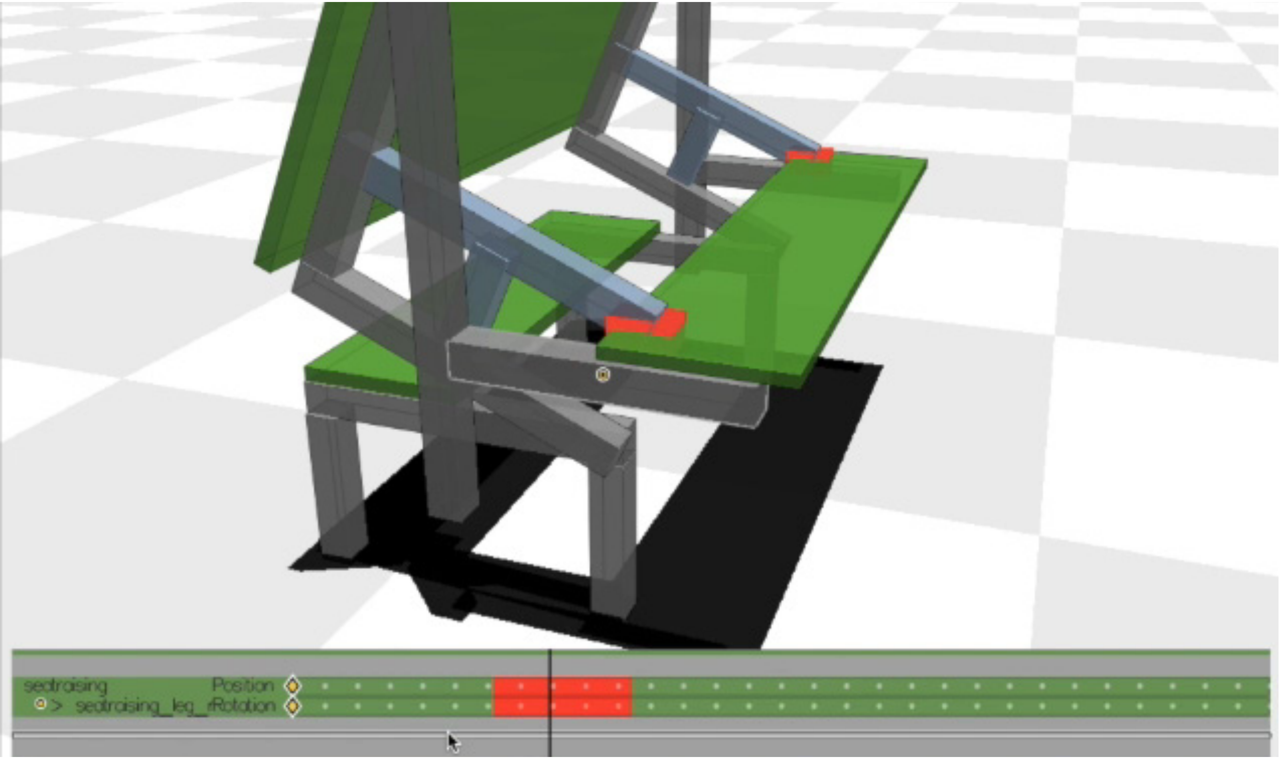
Applying the chain rule reveals UI couples surface positions across time

$$\underset{\mathrm{UI}}{\operatorname{argmin}} \int_0^1 \|I_C(t)\| \, dt$$

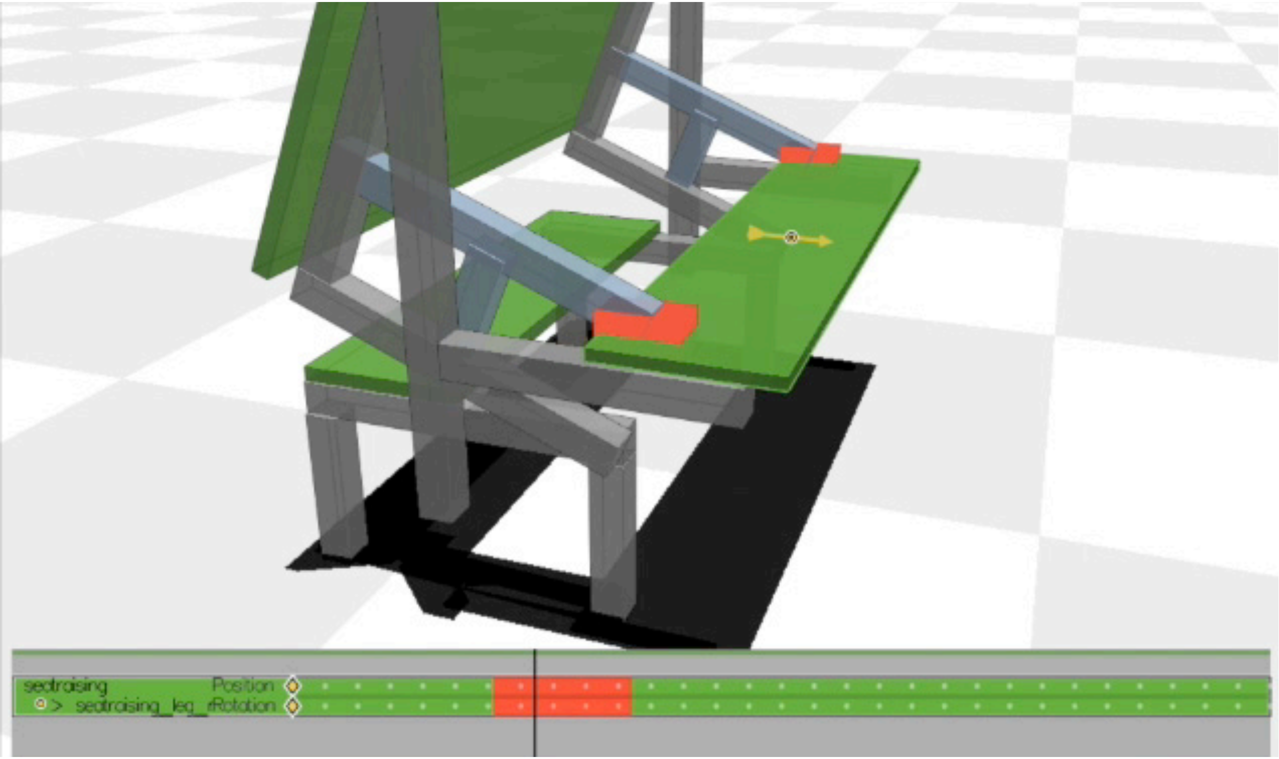
$$\mathrm{UI} \leftarrow \mathrm{UI} - \int_0^1 \boxed{(\nabla_{\mathrm{UI}} \mathbf{V}_t)} (\nabla_{\mathbf{V}_t} \|I_C(t)\|) \; dt$$

See paper for details…
[Harmon et al. 2011]
[Umetani et al. 2011]

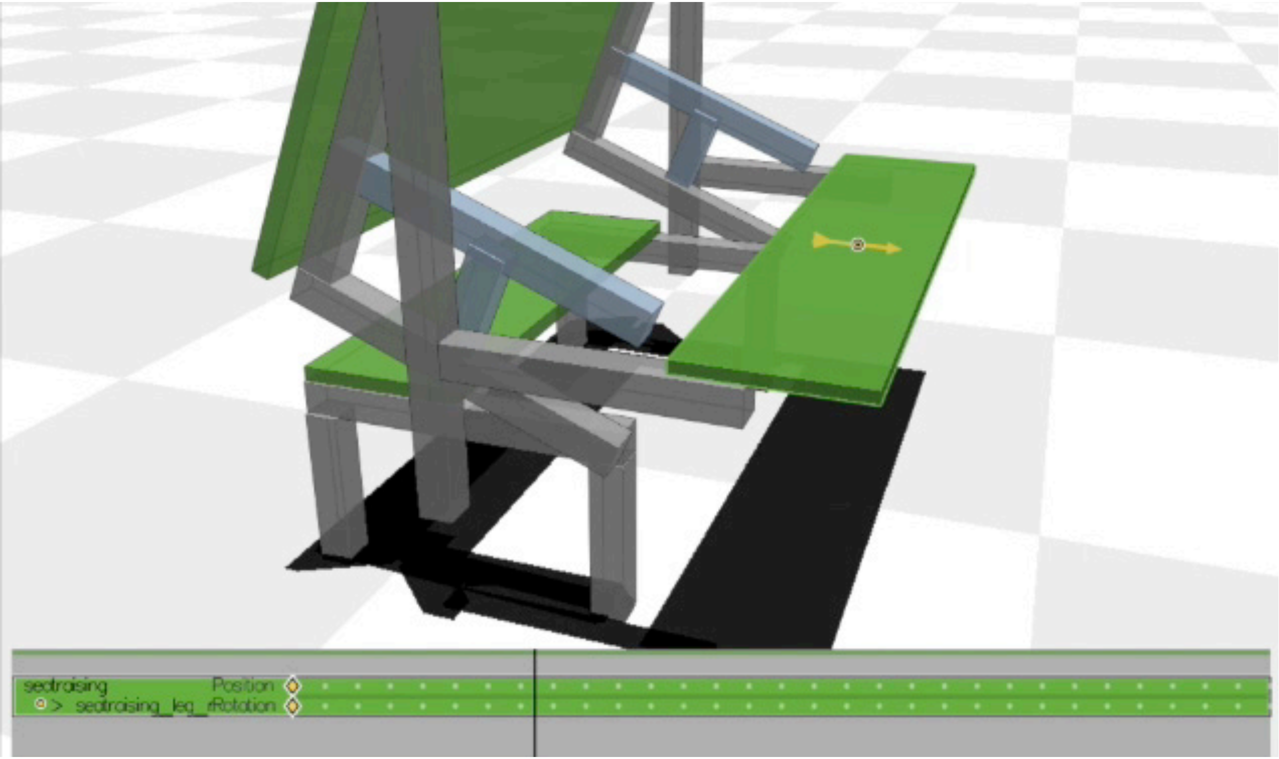
Gradient direction with respect to UI immediately helps as hint



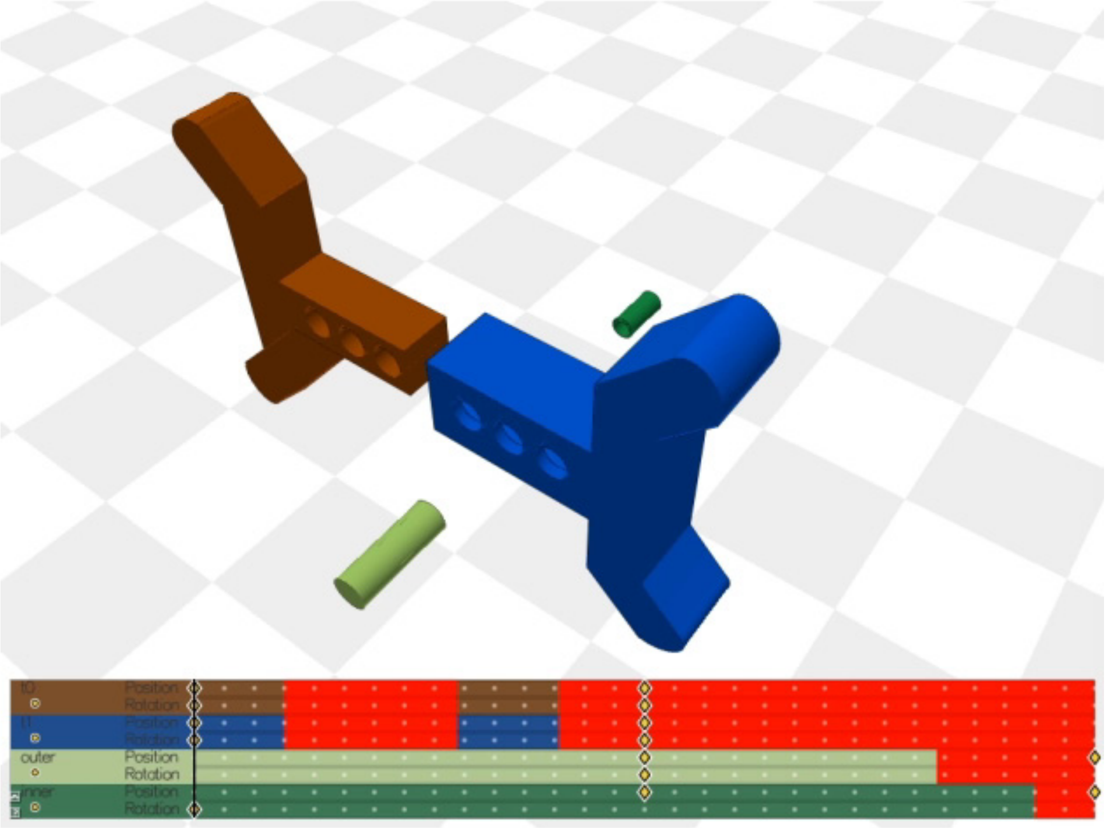
Gradient direction with respect to UI immediately helps as hint



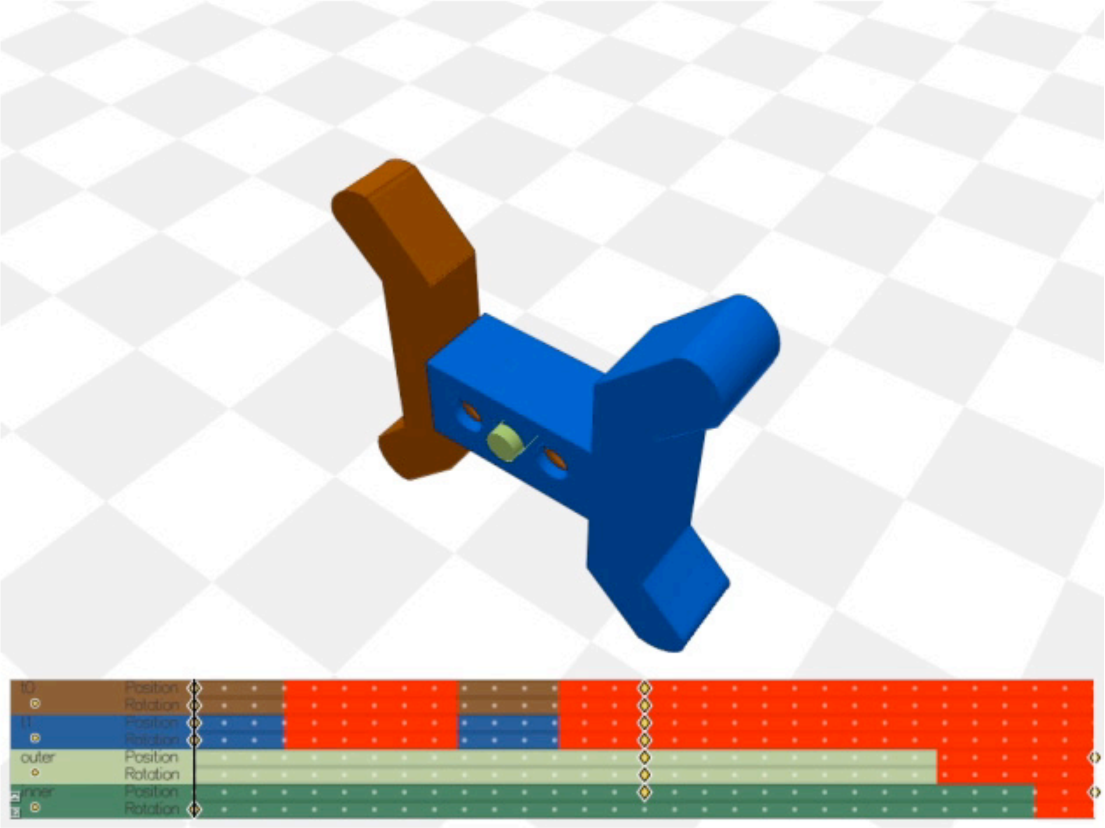
Gradient direction with respect to UI immediately helps as hint



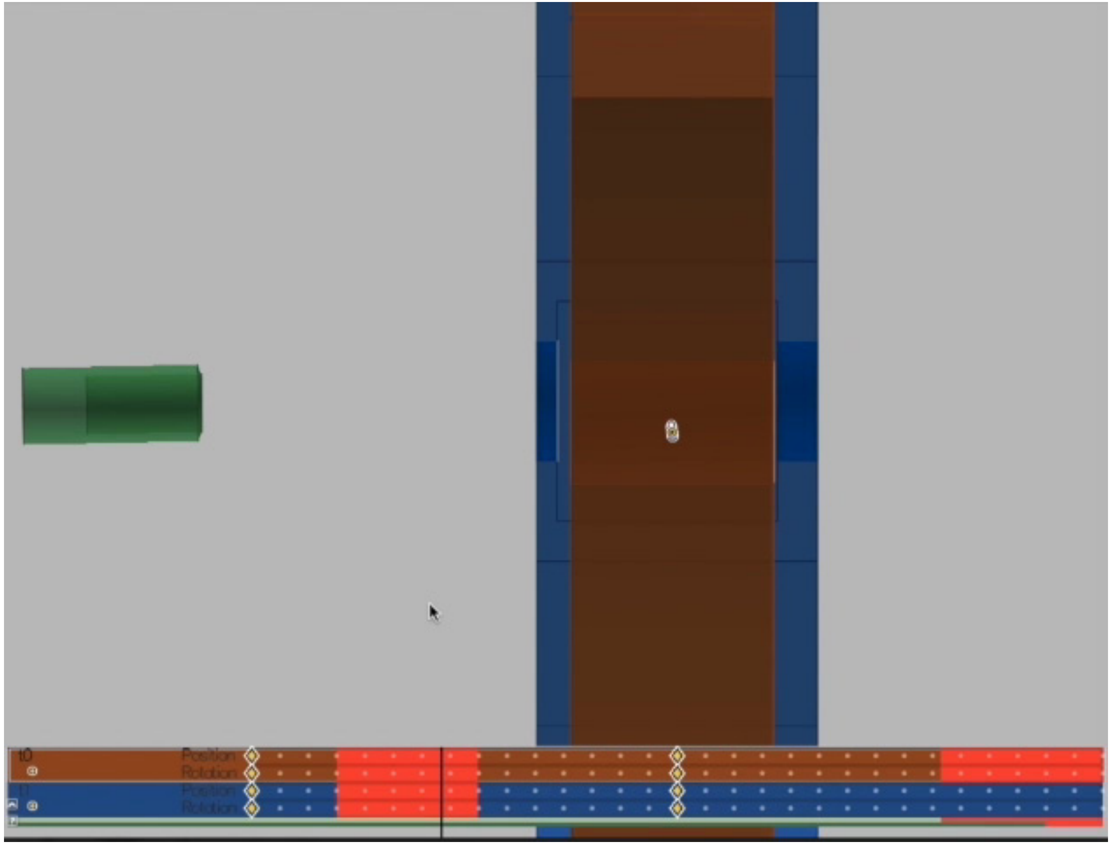
Collisions involving many objects can be very tedious to fix…



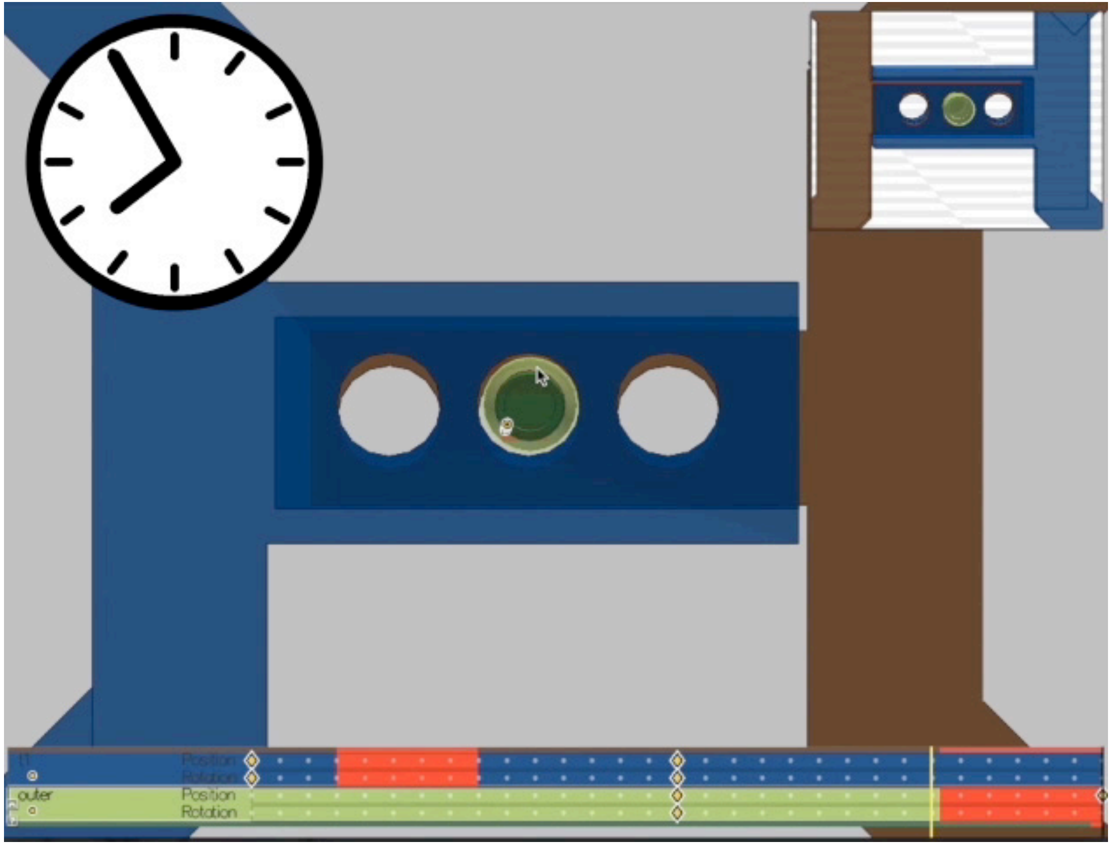
Collisions involving many objects can be very tedious to fix…



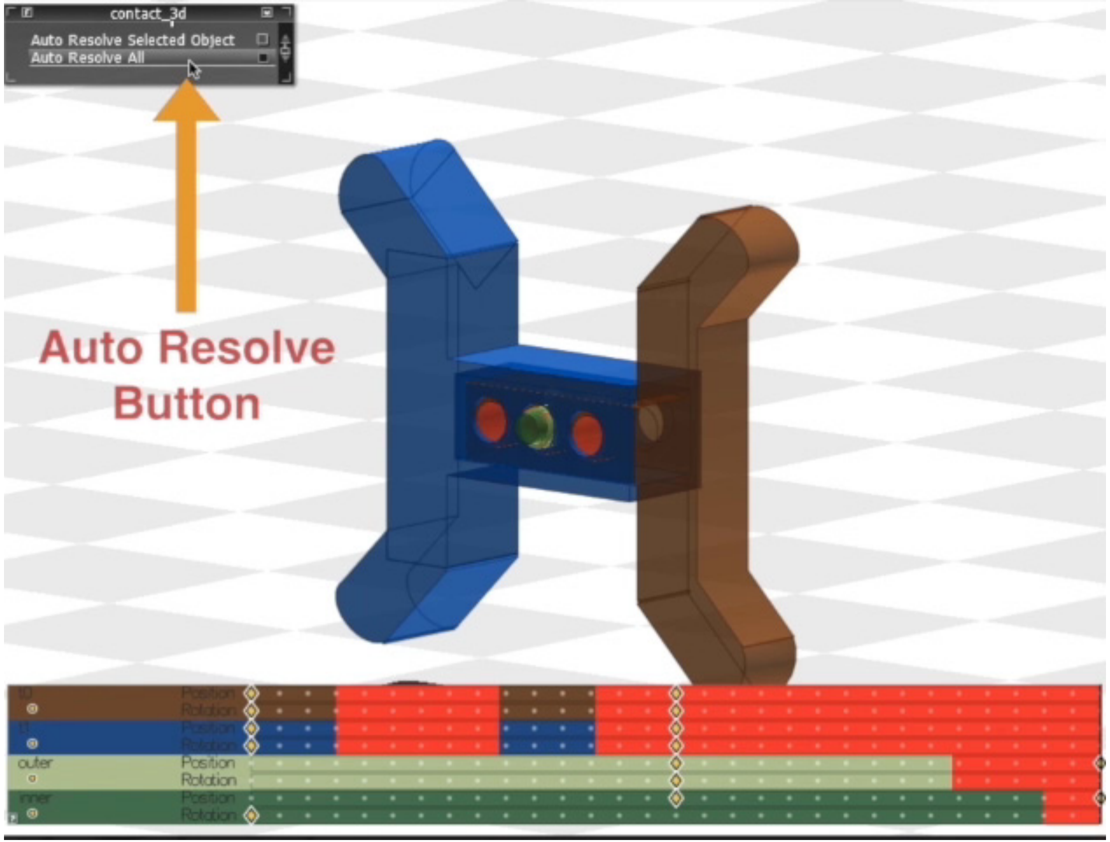
… manually fixing one collision may cause more at another place or time



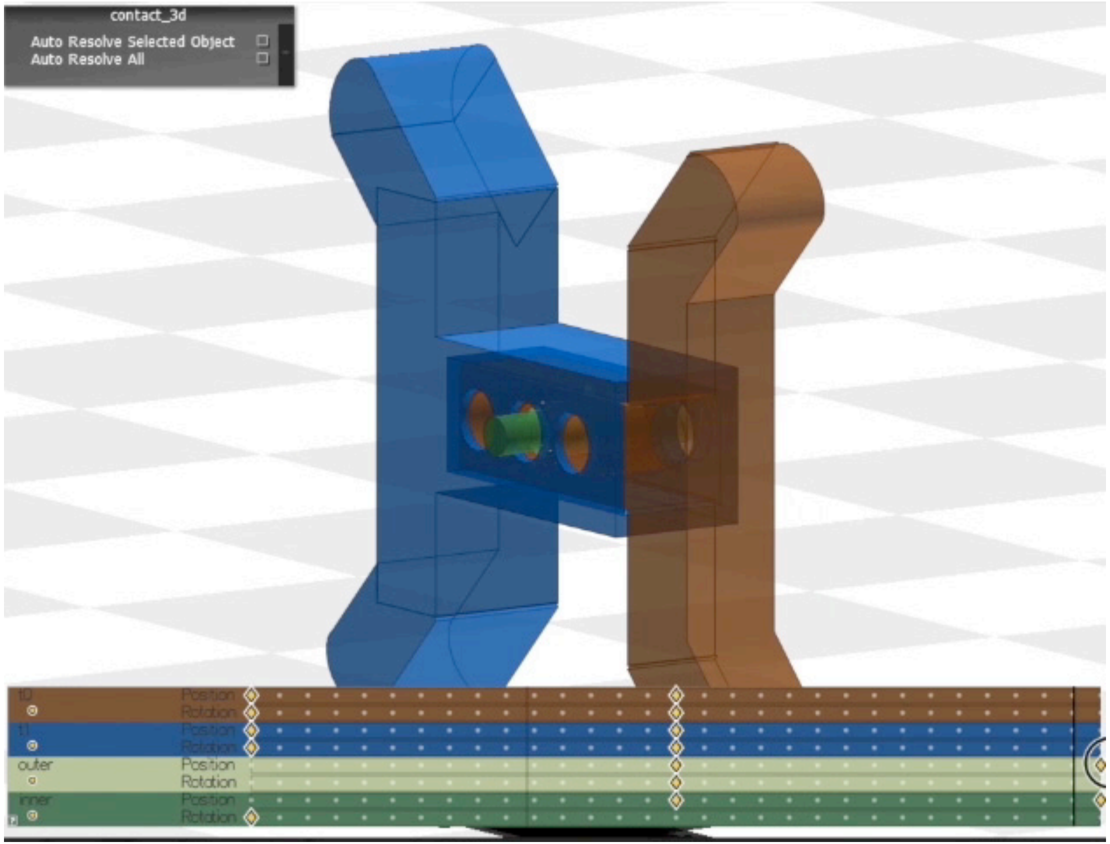
… manually fixing one collision may cause more at another place or time



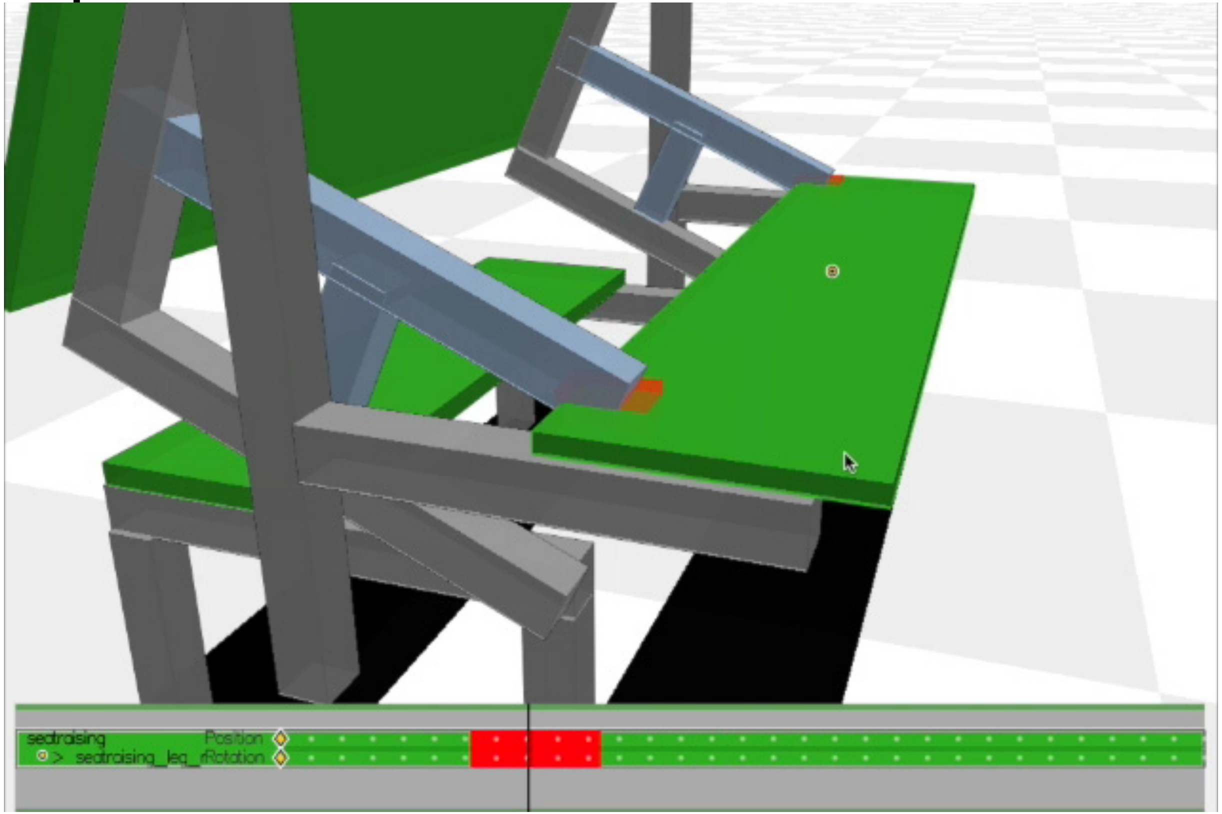
… instead we may follow gradient descent until all collisions disappear



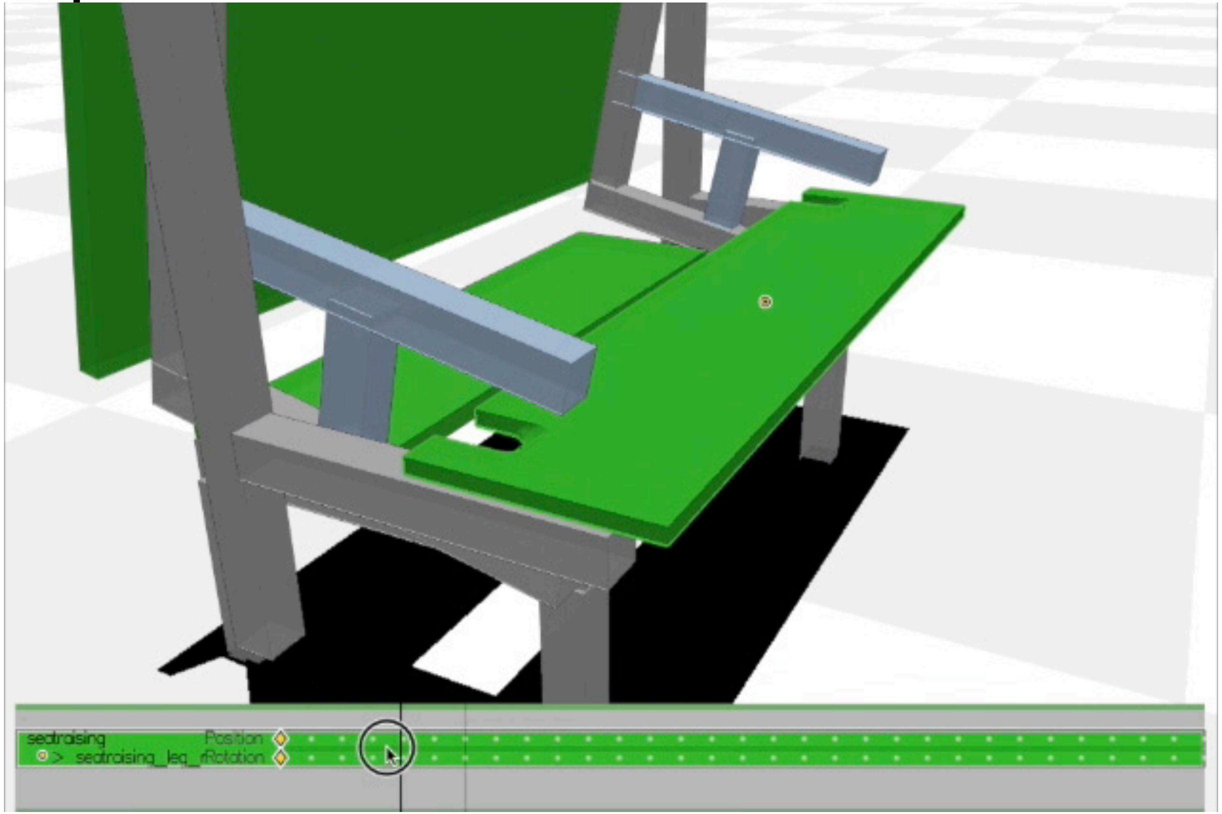
… instead we may follow gradient descent until all collisions disappear



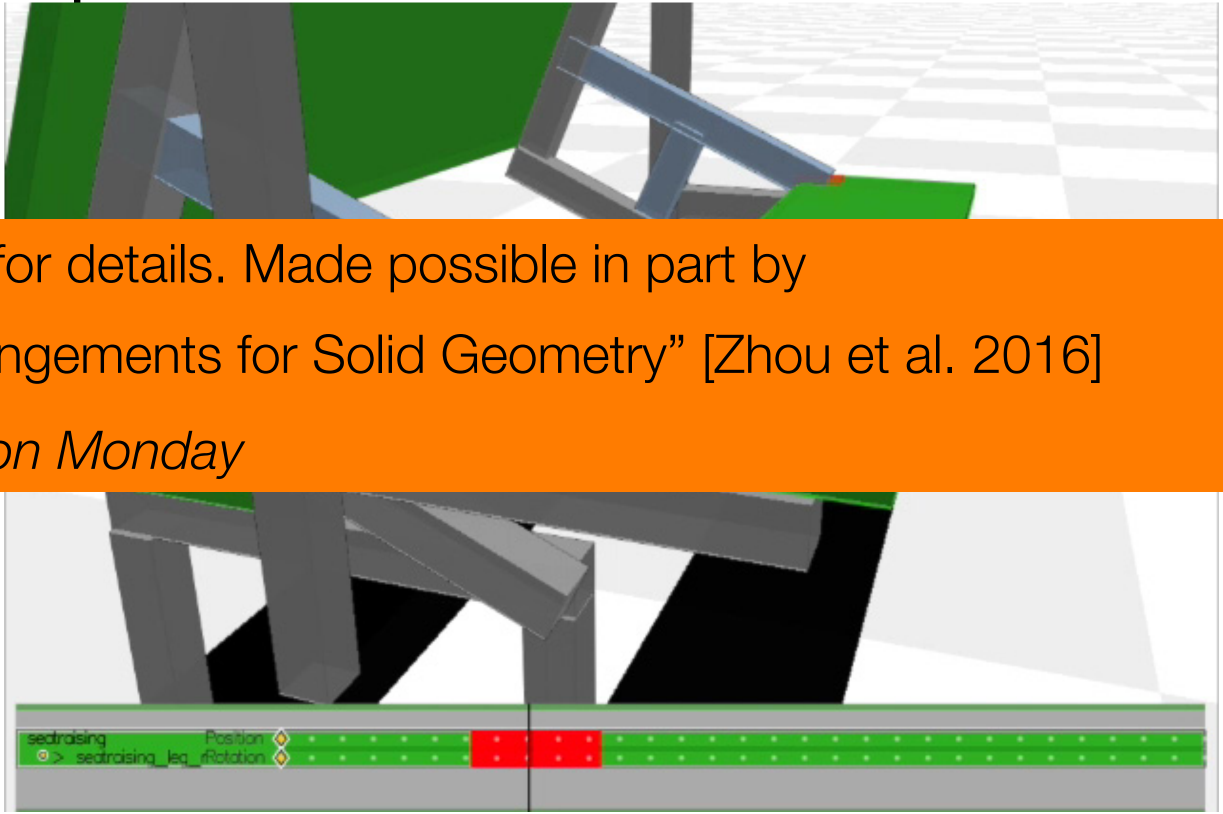
Sometimes, one part's geometry is less important than another's…



Sometimes, one part's geometry is less important than another's…



Sometimes, one part's geometry is less important than another's…

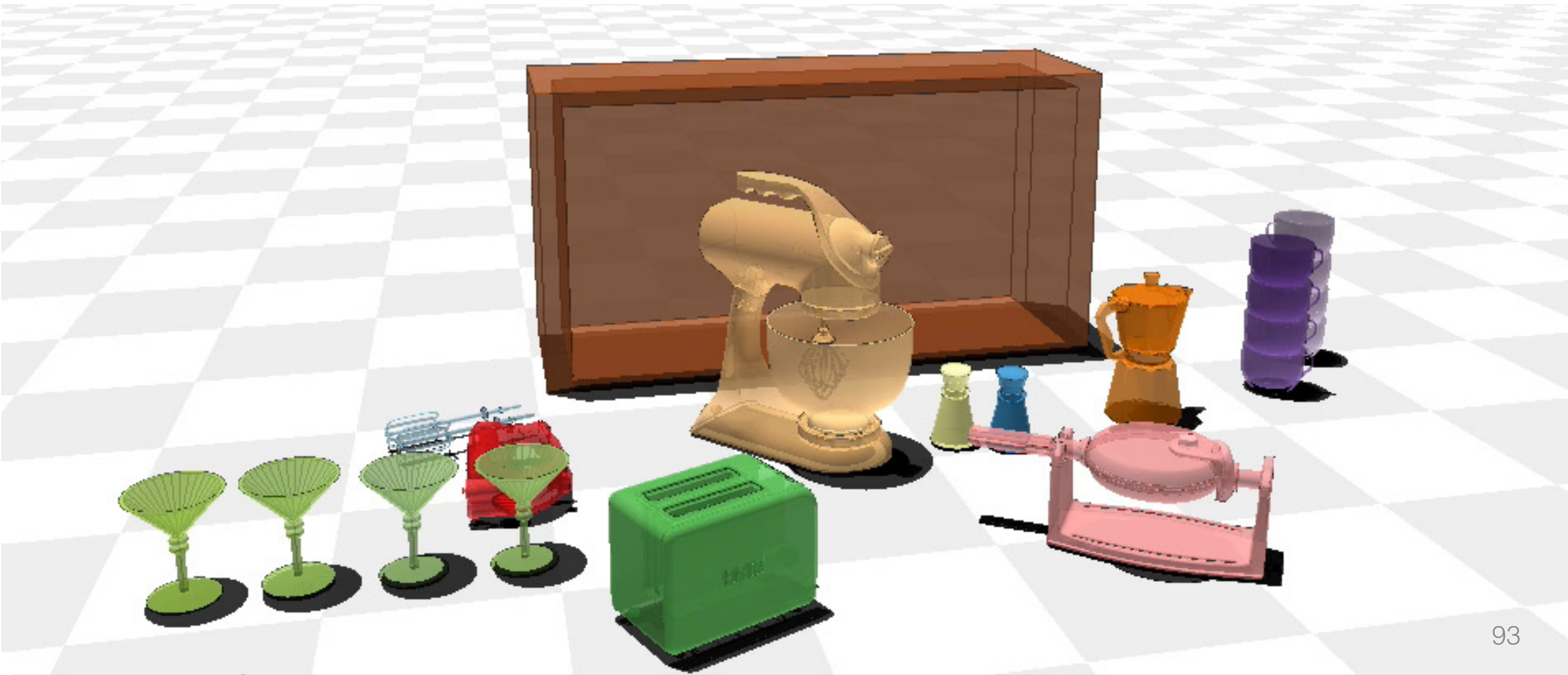


See paper for details. Made possible in part by

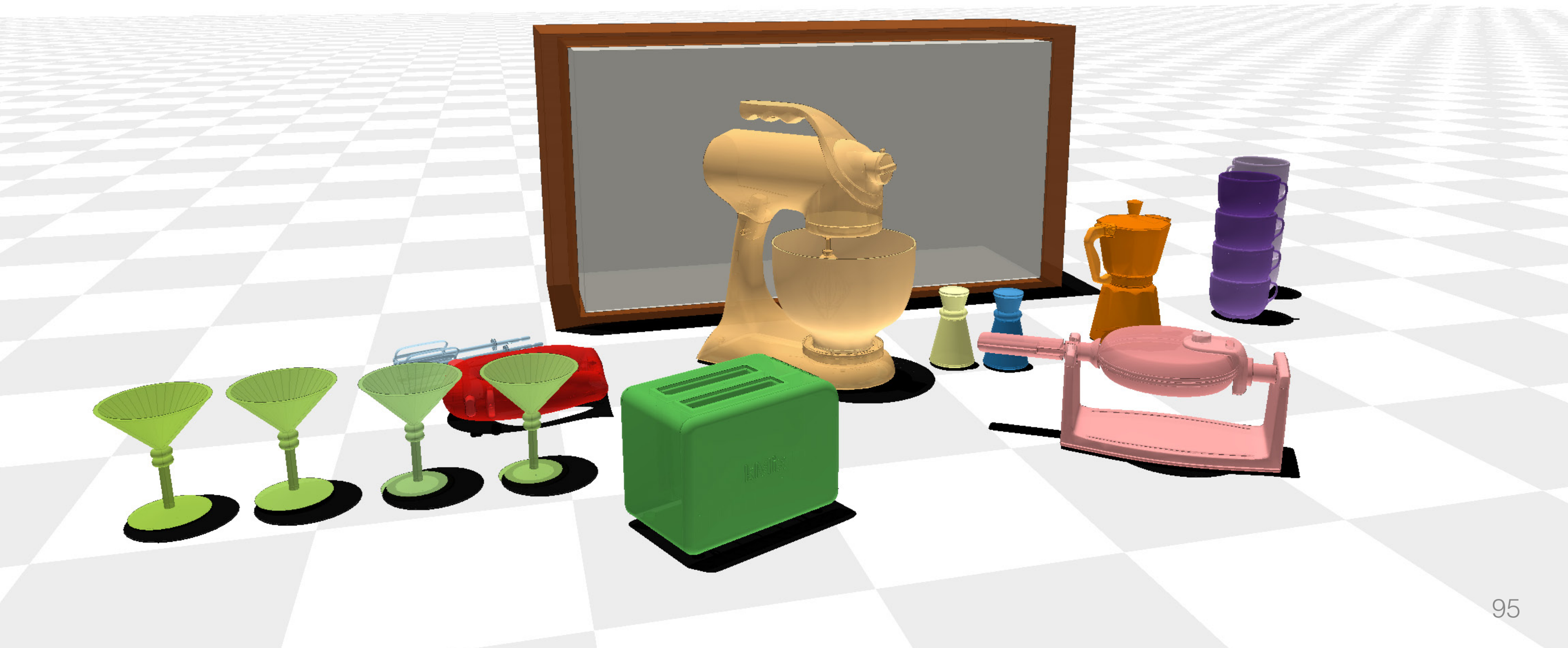
"Mesh Arrangements for Solid Geometry" [Zhou et al. 2016]

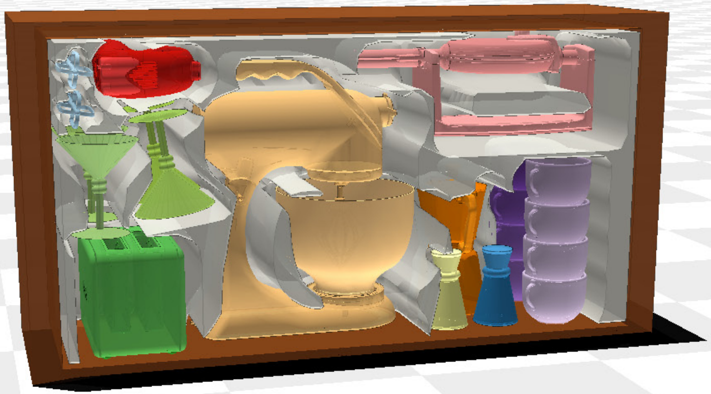
presented on Monday

Swept volume carving invites creative space-saving solutions



Swept volume carving invites creative space-saving solutions





Limitations & Future Work

modeling space extend to full modeling / deformation tools

Limitations & Future Work

modeling space	extend to full modeling / deformation tools
data	integrate with 3D scanning / model repos

Limitations & Future Work

modeling space	extend to full modeling / deformation tools
data	integrate with 3D scanning / model repos
physics	feasibility extends beyond collisions

Acknowledgements

We thank Keenan Crane and Henrique Maia for illuminating discussions.

The Columbia Computer Graphics Group is supported by Disney Research, Pixar, Adobe, and Altair.

Funded in part by NSF grants CMMI-11-29917, IIS-12-08153, IIS-14-09286, and IIS-17257.

Computational Design of Reconfigurables

Akash Garg

Alec Jacobson jacobson@cs.toronto.edu

Eitan Grinspun