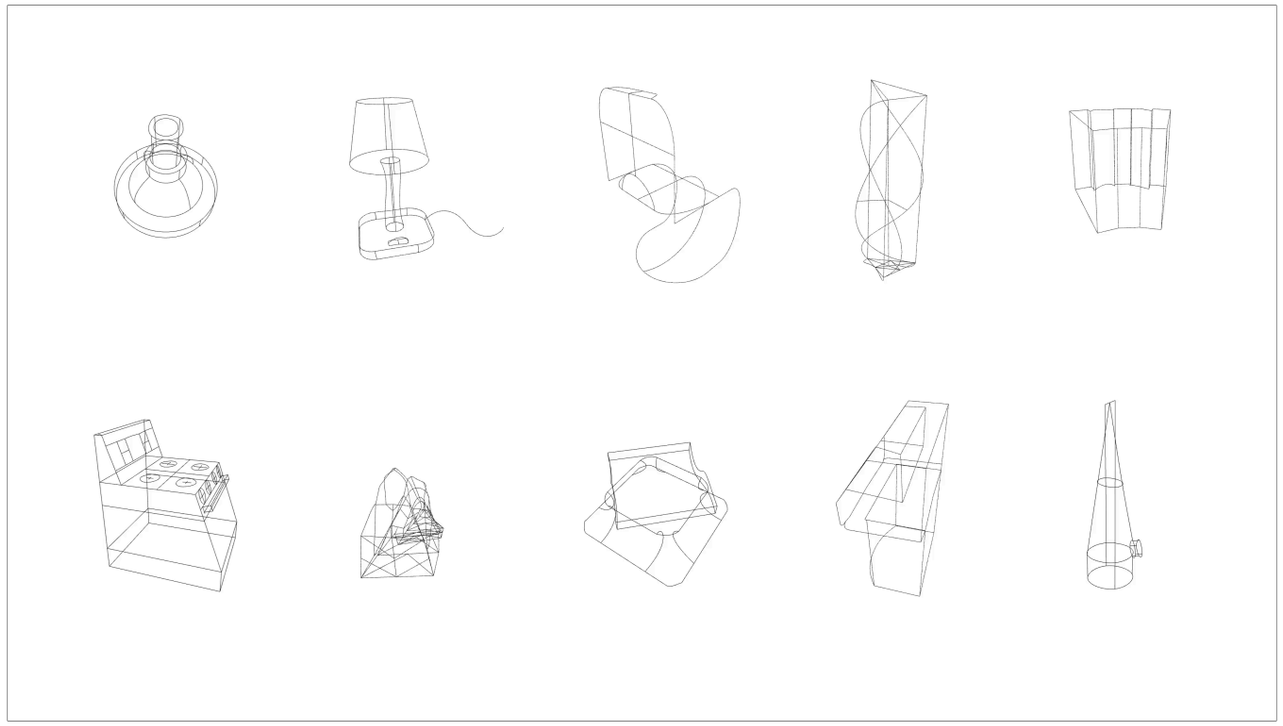
ScaffoldSketch

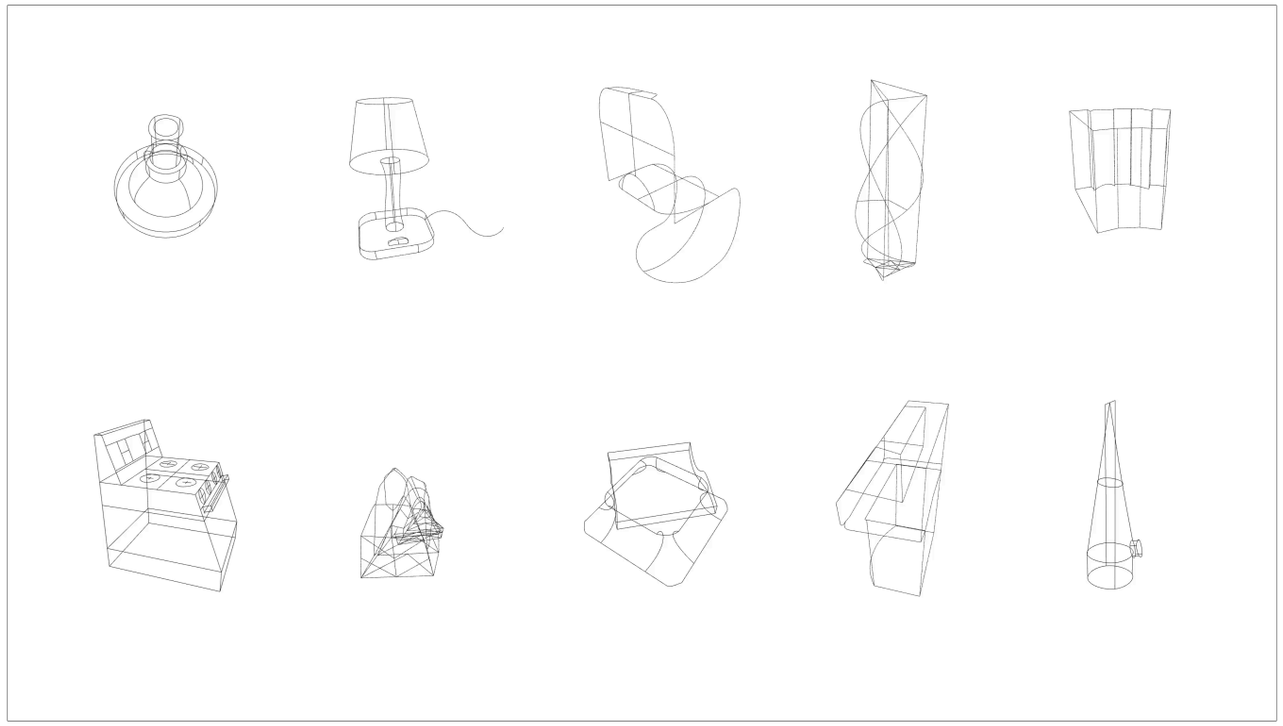
Accurate Industrial Design Drawing in VR

Xue Yu	George Mason University
Stephen DiVerdi	Adobe Research
Akshay Sharma	Virginia Tech, Iowa State University
Yotam Gingold	George Mason University

We introduce ScaffoldSketch



a system that makes it possible to draw accurate 3D models in VR.

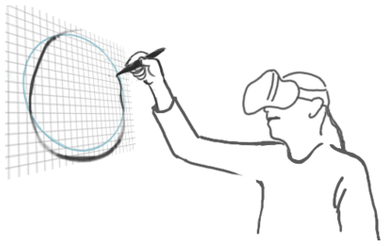


Drawing in VR is direct yet inaccurate

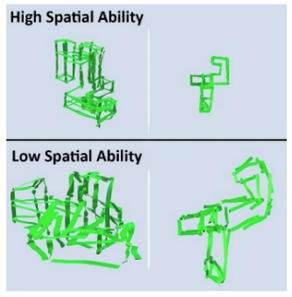
People like to draw in VR because it is immersive and the output is in 3D, but it’s really hard to be accurate.

Drawing in VR is direct yet inaccurate

Related Work: 3D Drawing Accuracy



[Arora, Kazi, Anderson, Grossman, Singh, and Fitzmaurice 2017]



[Machuca, Stuerzlinger, and Asente 2019]

...

Previous work has already proved this. More degrees of freedom, lack of physical surface, depth issue, these all contribute to larger errors directly drawing in VR.

Product designers draw accurately in 2D with scaffolds

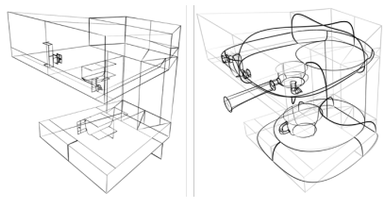
So how do people draw accurately in 2D?

Product designers can draw very accurate sketches.
When they draw with pen and paper, they'll first draw construction lines, or known as scaffolds.
Then they draw the object shape on top of the scaffolds.

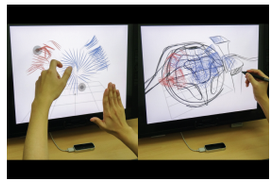
The scaffolds serve as visual guidance to make the shape strokes accurate.
This is how an industrial designer can draw accurate trash can.

Product designers draw accurately in 2D with scaffolds

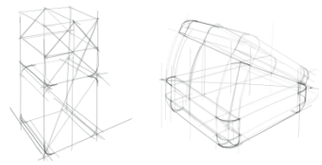
Related Work: Interpreting Scaffolds



[Schmidt, Khan, Singh, and Kurtenbach 2009]



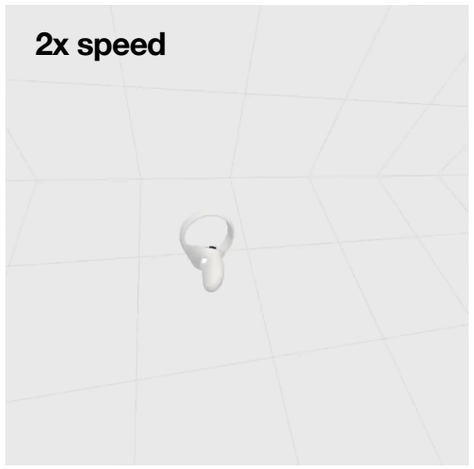
[Kim, An, Singh, Lee, and Bae 2018]



[Gryaditskaya, Sypesteyn, Hoftijzer, Pont, Durand, and Bousseau 2019]

There are also some previous work studying/utilizing this scaffold plus shape way of sketching.

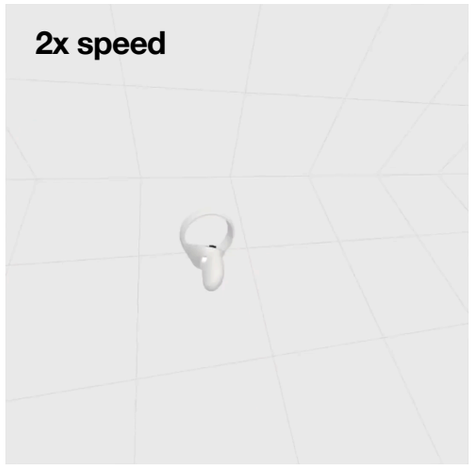
Drawing scaffolds in VR doesn’t solve accuracy



Can we borrow this approach directly in VR?

Even with the help of scaffolds, people still cannot draw accurately. The reason is similar as before, with more degrees of freedom it requires higher motion control.
Even the scaffolds are not perfect aligned with each other.
(We also did a controlled experiment, see our paper for details.)
If we could eliminate these issues we would have a nice way to draw accurately in VR.

Drawing scaffolds in VR doesn’t solve accuracy

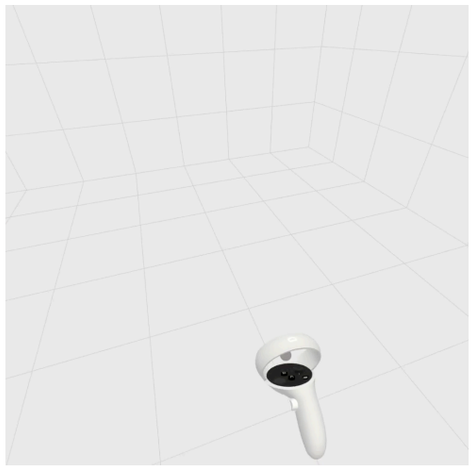


There is enough structure in scaffold and shape strokes to auto-correct

We introduced ScaffoldSketch. A system helps people draw accurately in VR by auto-correcting scaffolds and stroke strokes.

There is enough structure in scaffold and shape strokes to auto-correct

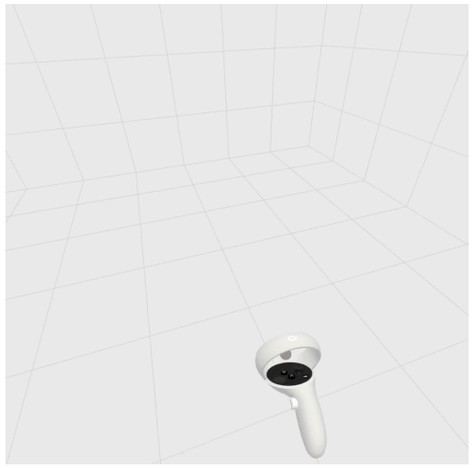
Scaffolds are straight lines with special relationships



- Attach at endpoints
- Parallel
- Perpendicular
- Same Length

Scaffolds are straight lines with special relationships. So they can snap to each other based on endpoints, directions and length.
We also support tick on the scaffold lines to mark intermediate positions as designers often do with pen and paper.

Scaffolds are straight lines with special relationships



- Attach at endpoints
- Parallel
- Perpendicular
- Same Length

Auto-Correcting Scaffolds

Here’s an example with black lines as existing scaffold lines.
When the user draw a new scaffold line.

We can consider these relationships with existing scaffolds.
Relationships may conflict with each other,Previous attempts to resolve conflicts often use heuristics or don’t satisfy any;

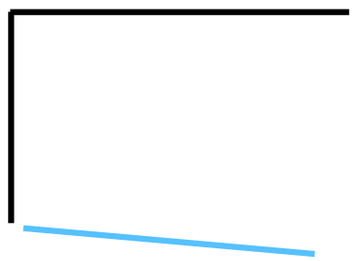
==========
The relationships we consider are: Endpoints snapping, parallel ,perpendicular for directions and same length.
For all the existing scaffold lines, we consider their endpoints, midpoints and the user defined intermediate positions as key points.

And whenever we have a newly drawn scaffold line, we’ll check whether the newline’s endpoints near any key points. If so, we’ll try to snap to them. And this is a point-point relationship.

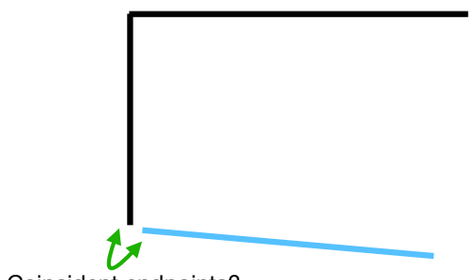
If the newline’s endpoints are close to existing lines, we’ll try to snap to the existing line, and this a point-line relationship.
If the new line is almost perpendicular or parallel to any existing lines. These are perpendicular and parallel constraints.
And if the new line’s length is close to any existing scaffold lines, this is an equal length constraint.

We might get many constraints for the new line, and these constraints might conflicting with each other, we will prior the point-point constraints and use iteratively re-weighted least squares (IRLS) to converge to a set of non-conflicting constraints.

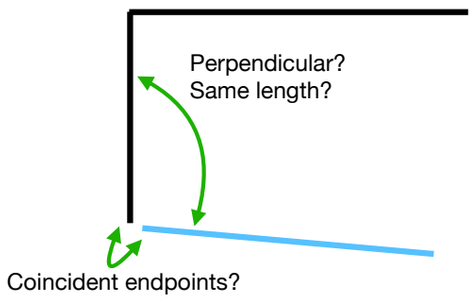
Auto-Correcting Scaffolds



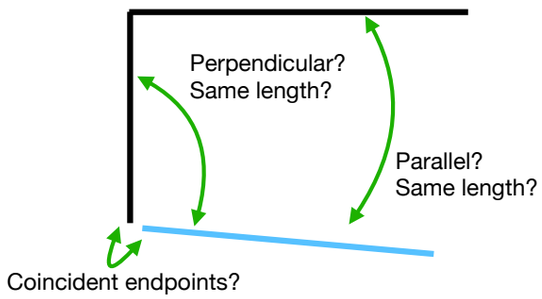
Auto-Correcting Scaffolds



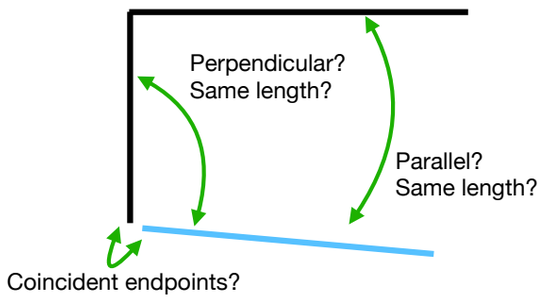
Auto-Correcting Scaffolds



Auto-Correcting Scaffolds

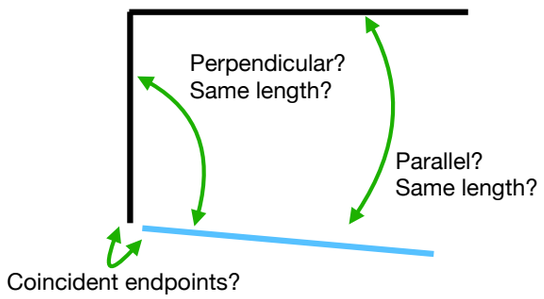


Auto-Correcting Scaffolds



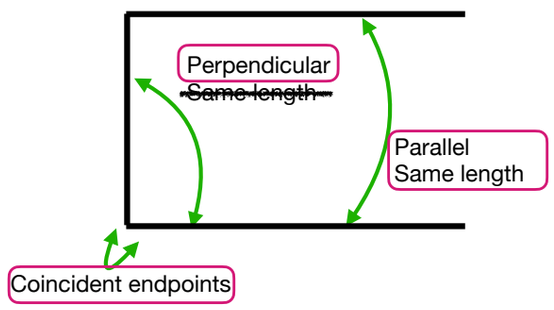
- Relationships may conflict with each other.

Auto-Correcting Scaffolds



- Relationships may conflict with each other.
- Previous attempts to resolve conflicts often use heuristics or don’t satisfy any.

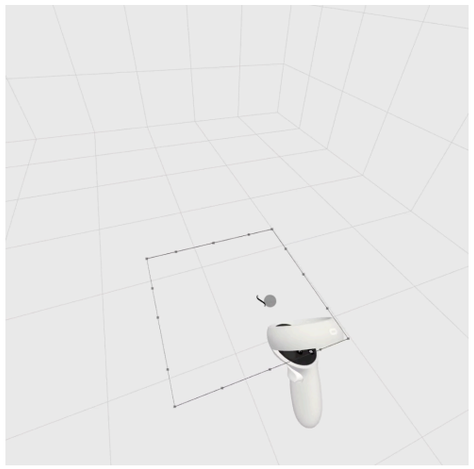
Auto-Correcting Scaffolds



- Relationships may conflict with each other.
- Previous attempts to resolve conflicts often use heuristics or don’t satisfy any.
- We resolve with an iteratively re-weighted least squares scheme.

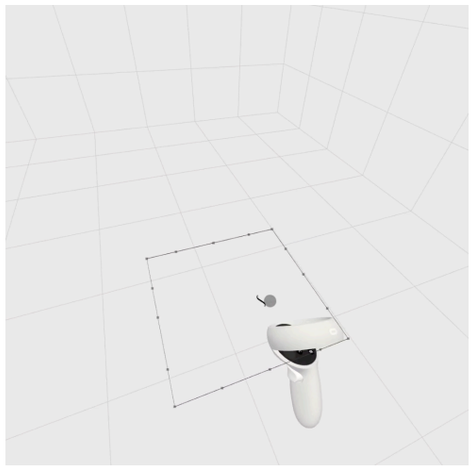
We resolve with an iteratively reweighed least square scheme. The weights will be updated after each iteration, the more satisfiable constraints will be given larger weights and the less satisfiable will have smaller weights. In the end the weights will be either infinity or 0, that is satisfied or discard for the constraints.

Shape strokes attach to scaffolds

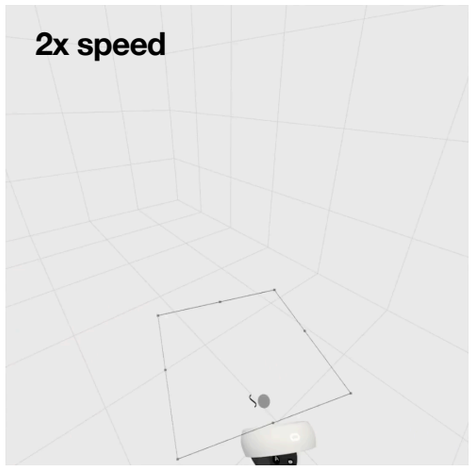


The shape strokes are smooth curves or lines attach to scaffolds, mainly attached to the endpoints, midpoints or the user marked positions, and usually tangent to those points.

Shape strokes attach to scaffolds



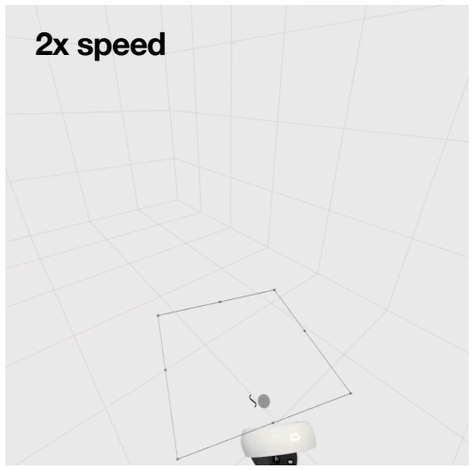
Same scaffold, different shape strokes



For the same scaffold square, different user input will be auto-corrected to different results.

The shape strokes are auto-corrected based on the scaffolds and the users drawing, just like how the scaffolds support shape strokes in 2D.

Same scaffold, different shape strokes

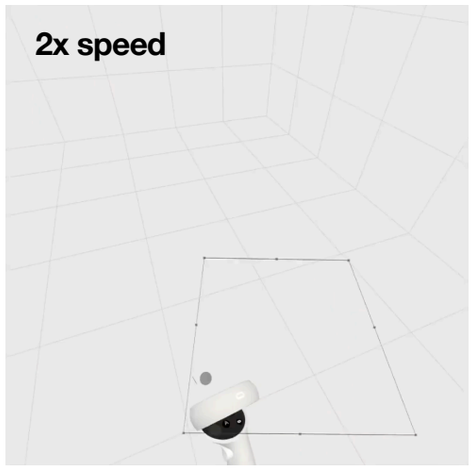


Space curves

We also support space curves as shown here.

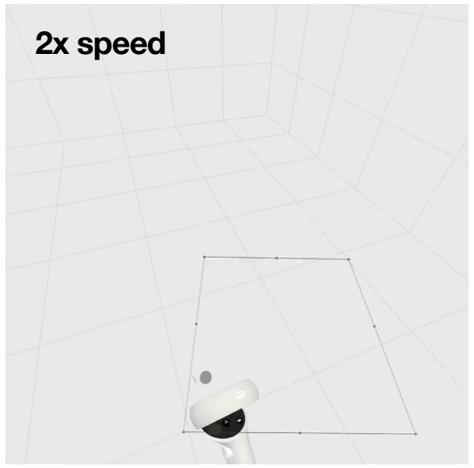
Space curves

Tick marks

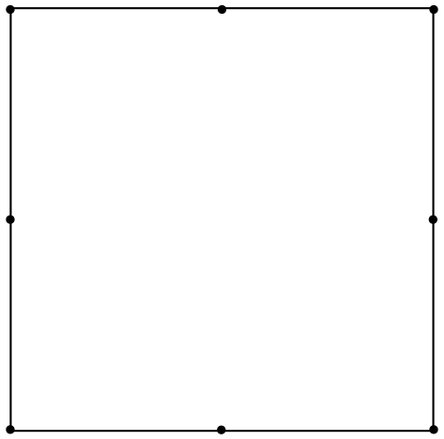


And with more tick marks as intermediate positions the auto-correct result will also be different.

Tick marks



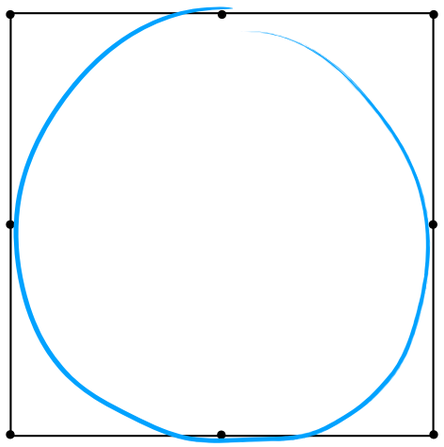
Auto-Correcting Shapes



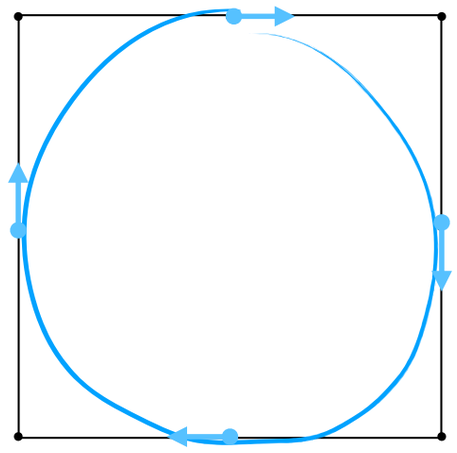
Once we have the beautified scaffolds, now if the user draw a shape curve, we can find whether curve is tangent to the intersections, thus we could have the curve directions and positions at those key points.

Alternate title: Shape Auto-Correct relies on key points and tangents

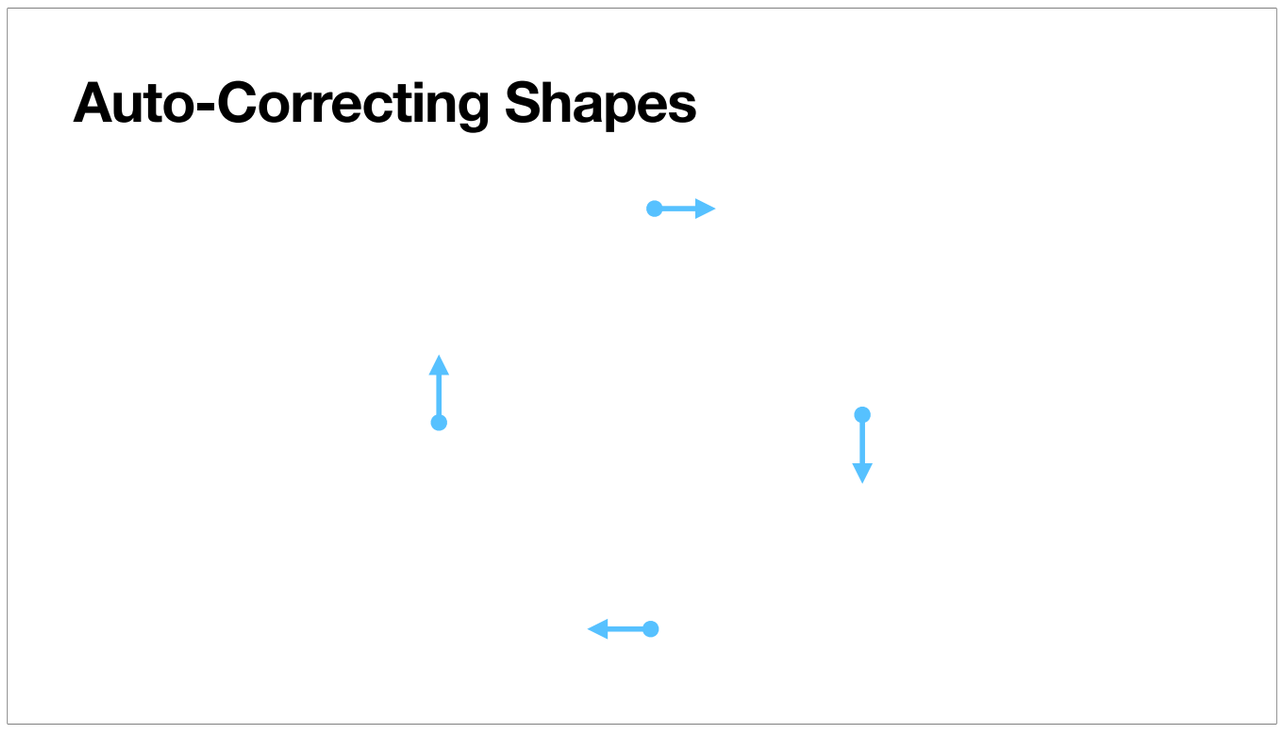
Auto-Correcting Shapes



Auto-Correcting Shapes



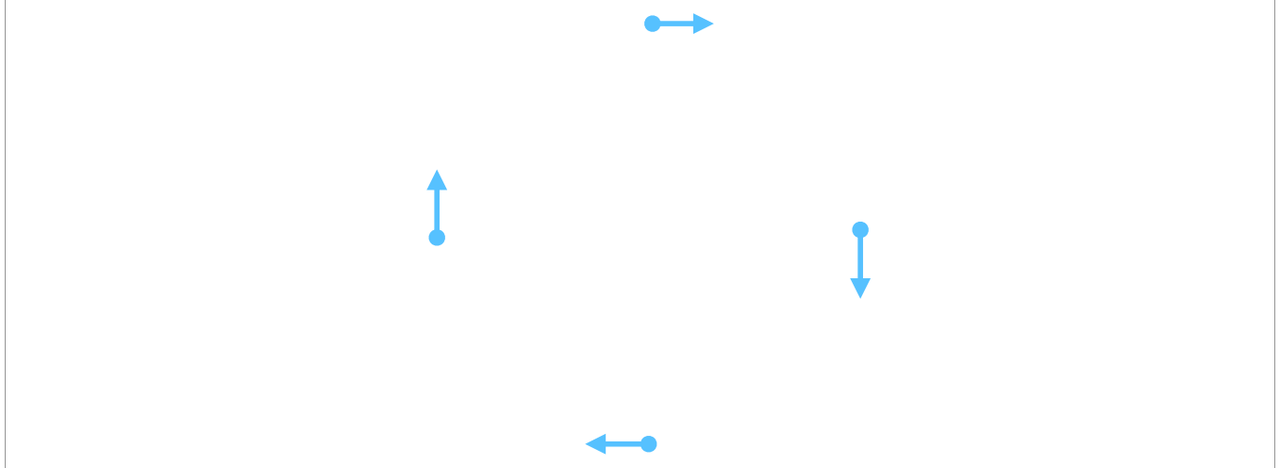
Auto-Correcting Shapes



The positions and directions are fixed at those points, but we change the magnitude of the tangents.

Alternate title: Shape Auto-Correct relies on key points and tangents

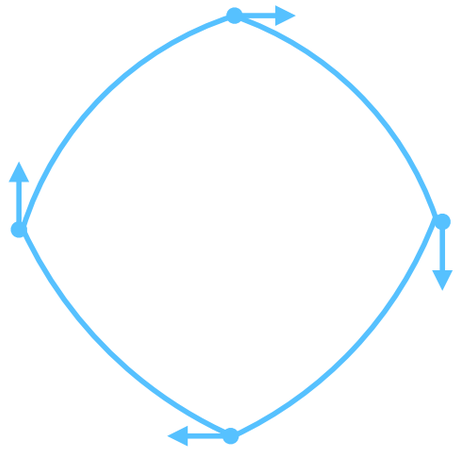
Auto-Correcting Shapes



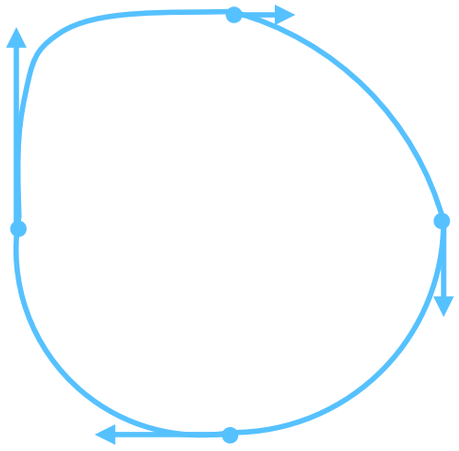
Different magnitudes of tangents will give us different curves.

Alternate title: Shape Auto-Correct relies on key points and tangents

Auto-Correcting Shapes

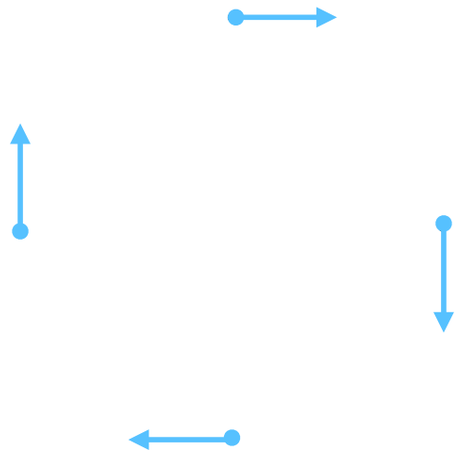


Auto-Correcting Shapes



Alternate title: Shape Auto-Correct relies on key points and tangents

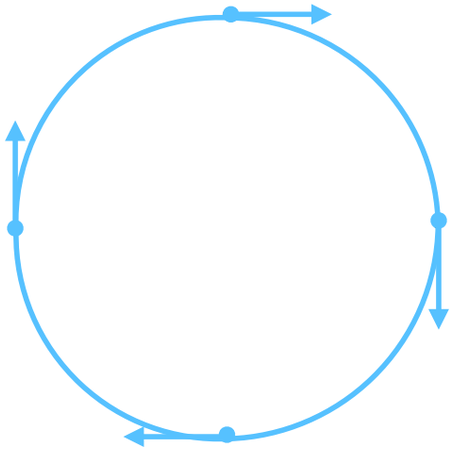
Auto-Correcting Shapes



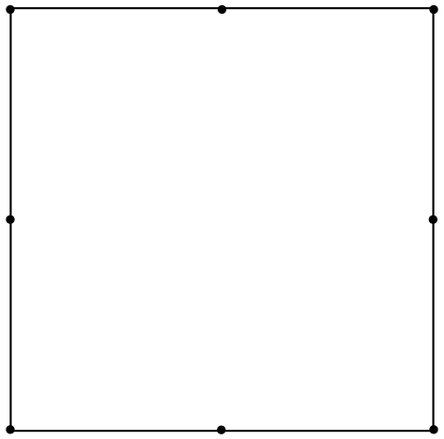
And our goal is to find the most smooth curve, that is the one of minimum variation of curvature.

Alternate title: Shape Auto-Correct relies on key points and tangents

Auto-Correcting Shapes



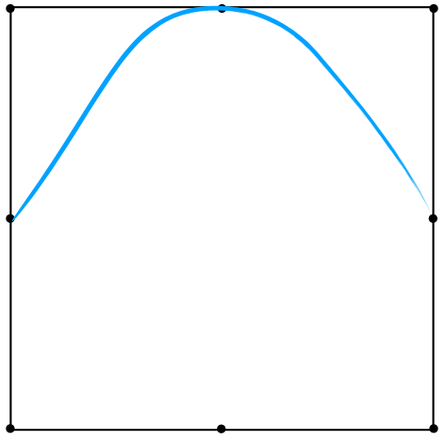
Auto-Correcting Shapes



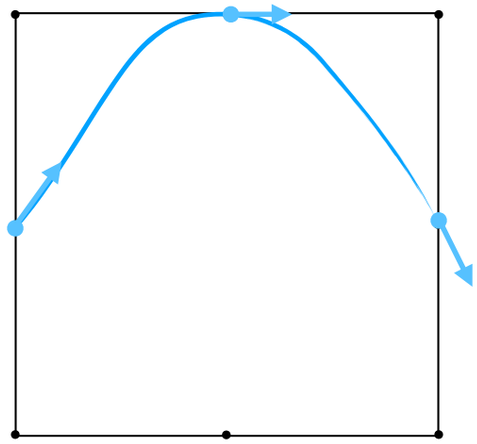
If the curve is not tangent to the scaffolds. we'll take the directions from original curve.

Alternate title: Shape Auto-Correct relies on key points and tangents

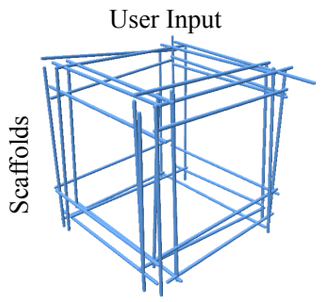
Auto-Correcting Shapes



Auto-Correcting Shapes



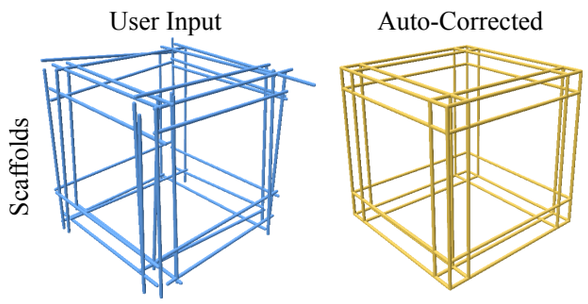
Example



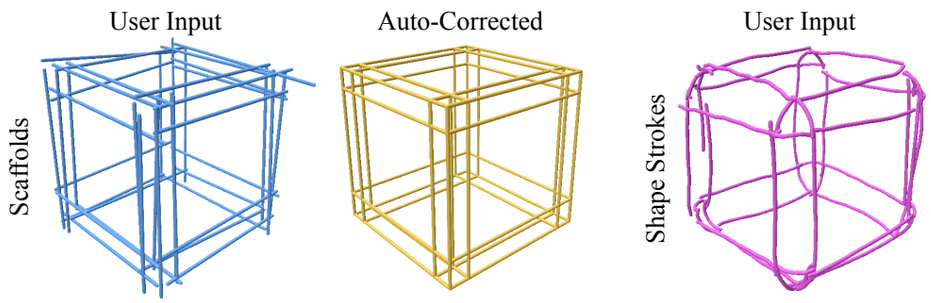
Here is the example of input scaffold. And the auto-corrected result.

And the shape strokes are based on the scaffolds. And the corresponding auto-corrected result.

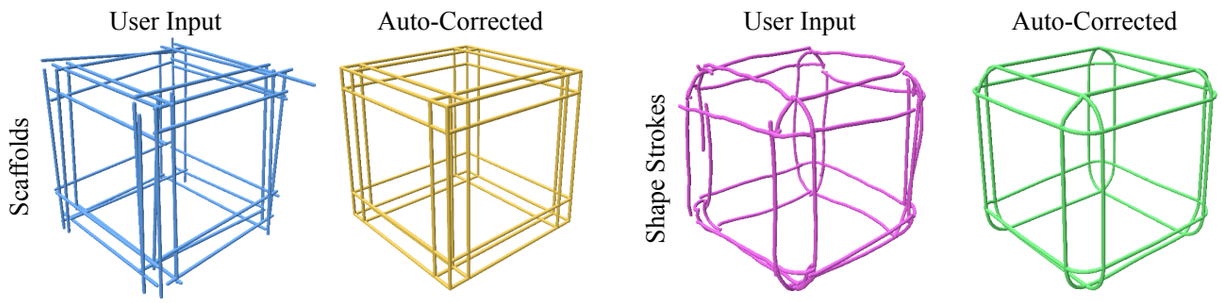
Example



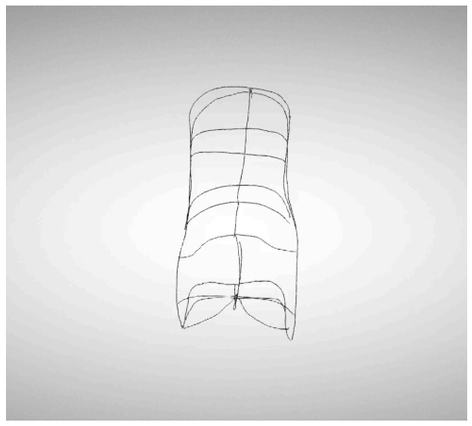
Example


User Input
Auto-Corrected
User Input
Scaffolds
Shape Strokes

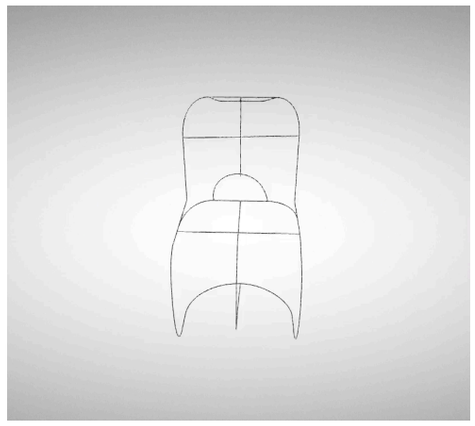
Example


User Input
Auto-Corrected
User Input
Auto-Corrected
Scaffolds
Shape Strokes

User Study



Without Auto-Correct



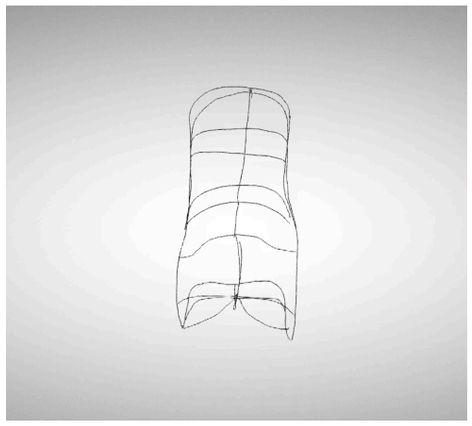
With Auto-Correct

We did a remote user study with 6 participants, all industrial design students.

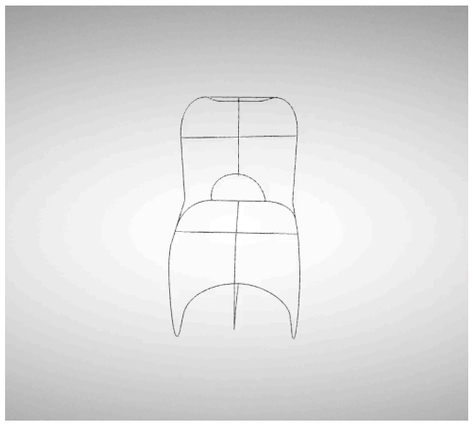
They used our system in two conditions: with and without auto-correct. They all expressed that auto-correct helped them draw more accurately, and they want to use ScaffoldSketch in the future. See the paper for details.

~~For more user study results and implementation details, please see our paper.~~

User Study

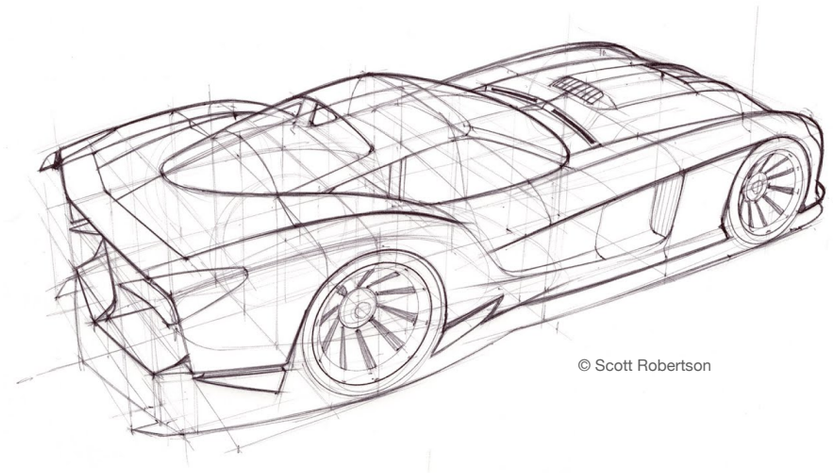


Without Auto-Correct



With Auto-Correct

Limitations and Future Work

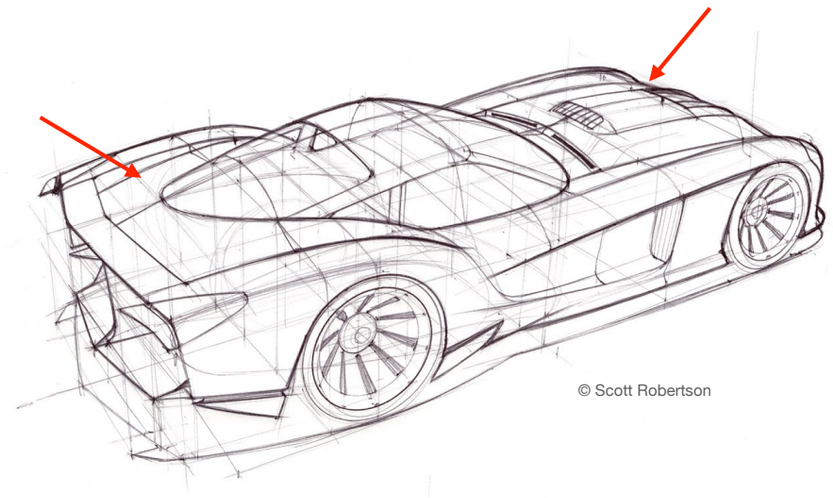


The current limitation and possible future work is - surfacing. We didn’t consider surfacing, which is also very important. Without surfacing, it’s hard to draw details and shading strokes.

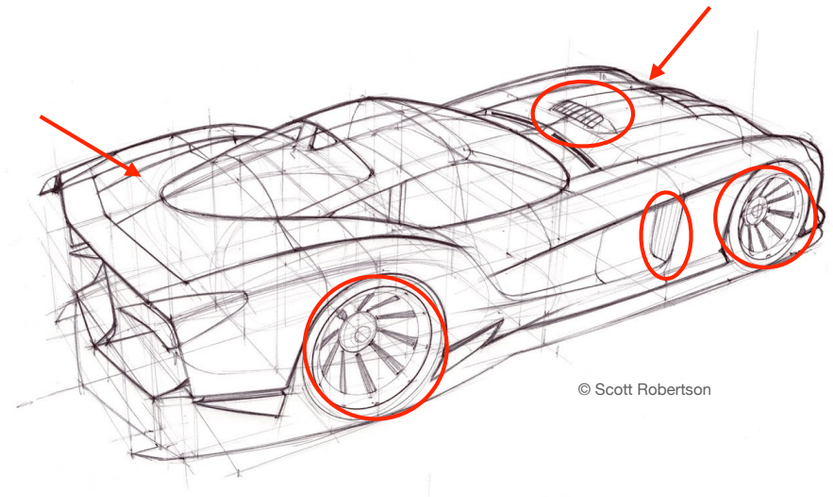
And since currently shape strokes are guided by the scaffolds, ScaffoldSketch don’t support free-form strokes.

With all these features combined, it will make a better user experience and more vivid 3D models.

Limitations and Future Work

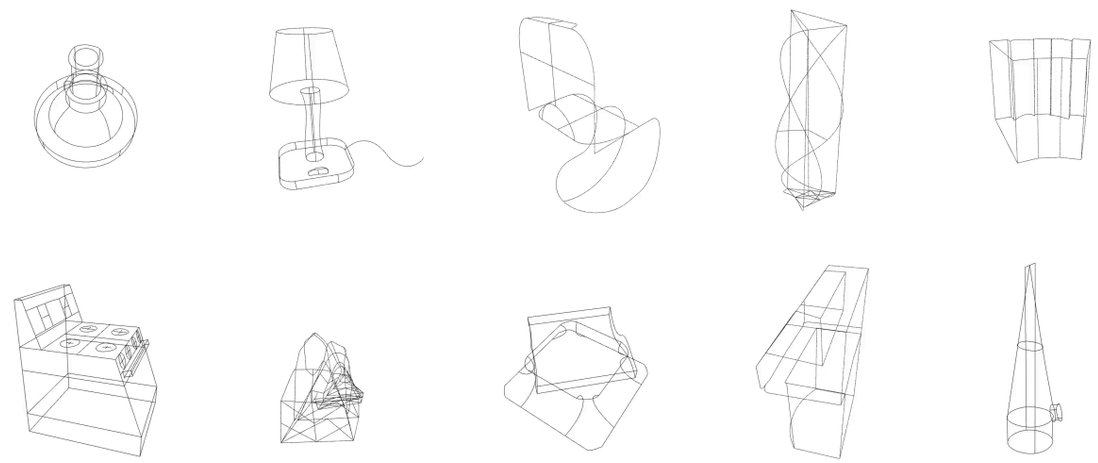


Limitations and Future Work



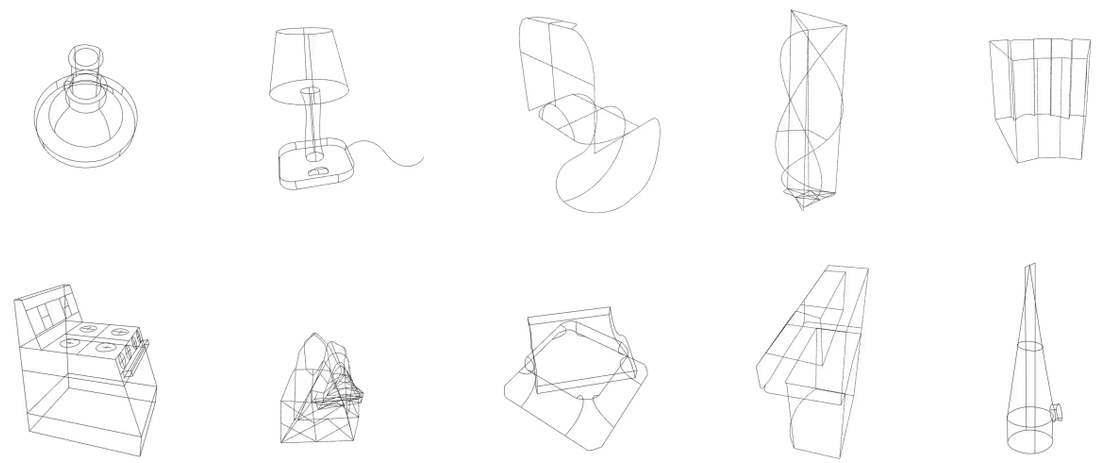
Conclusion

- ScaffoldSketch…
  - … enables precise 3D curve drawing by decomposing auto-correct into scaffold and shape stages.
  - … lets people transfer industrial design drawing skills into 3D.
  - … resolves conflicting constraints with an iterative re-weighting scheme.



Thank you

a system that makes it possible to draw accurate 3D models in VR.



Thank you