3DCG Illustrations [2026/06/01]

Non-Research Tips for Information Science Researchers

情報科学研究補助技法

Yuki Koyama | httsp://koyama.xyz/ | https://non-research-tips.github.io/

Yuki Koyama
小山 裕己

https://koyama.xyz/

- 2017:
  Ph.D. from **UTokyo** (Computer Science)
- 2017—2025:
  Researcher (Senior Researcher; 2022–)
  at **AIST (産業技術総合研究所)**
- 2021—Current:
  Technical Adviser at **Graphinica, Inc.**
  (an anime production company using 3DCG)
- 2025—Current:
  Associate Professor
  at **UTokyo** (Precision Engineering)

産総研
国立研究開発法人 産業技術総合研究所

GRAPHINICA

東京大学
THE UNIVERSITY OF TOKYO

Motivation: Why 3DCG?

In general, good illustrations ...

1. help audience better understand your research 💡 , and
2. increase presentation appeal 💖 (→ buzz).

3DCG is (sometimes) an effective technique for these purposes.


https://www.youtube.com/watch?v=HfHC0wNYry8

1. Better Understanding 💡

- Processed faster by the brain, **making it easier for audiences to grasp information**
- Provides visual representations that **make complex ideas and data more accessible**

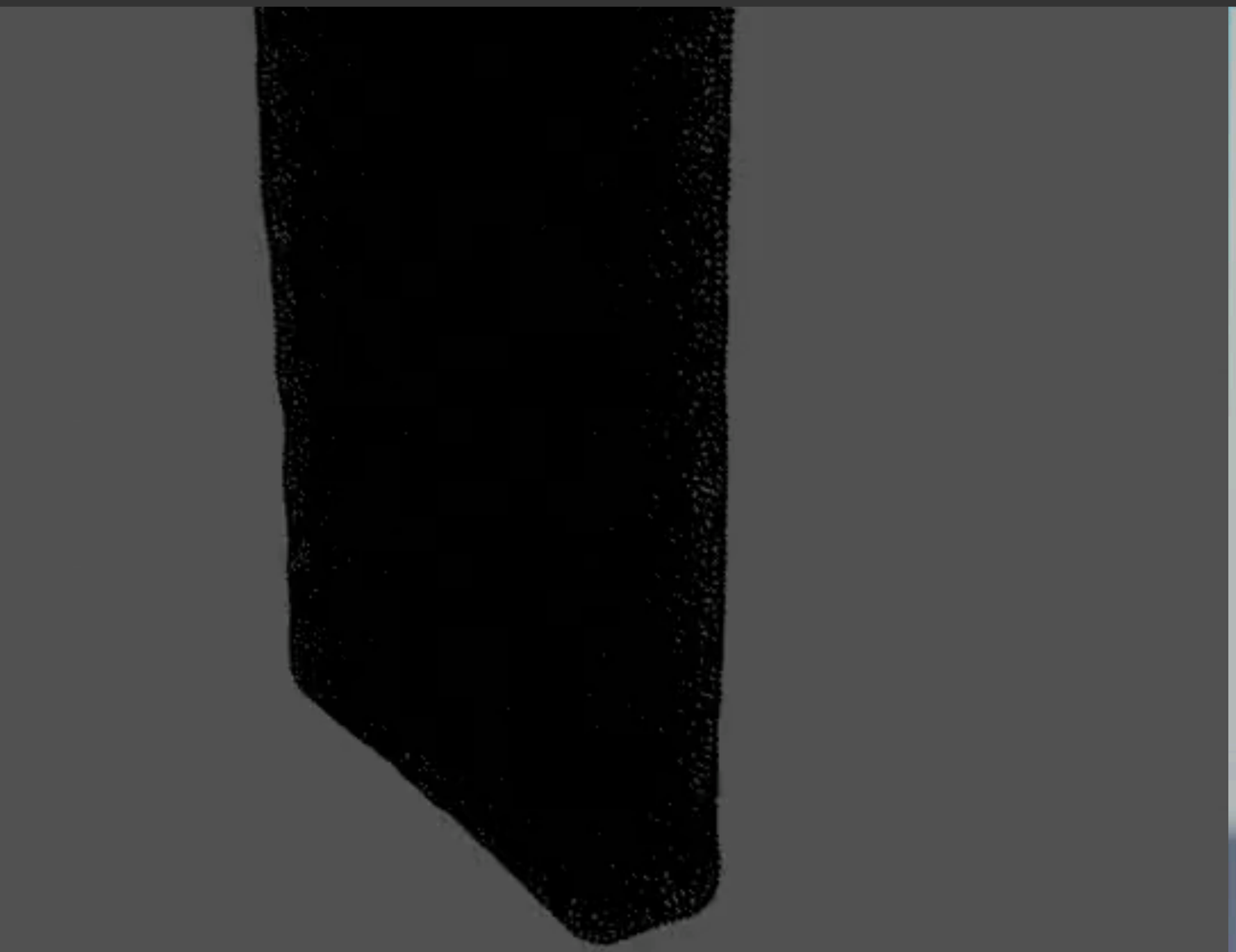
https://www.youtube.com/watch?v=2LPboySOSvo

2. Presentation Appeal 💖

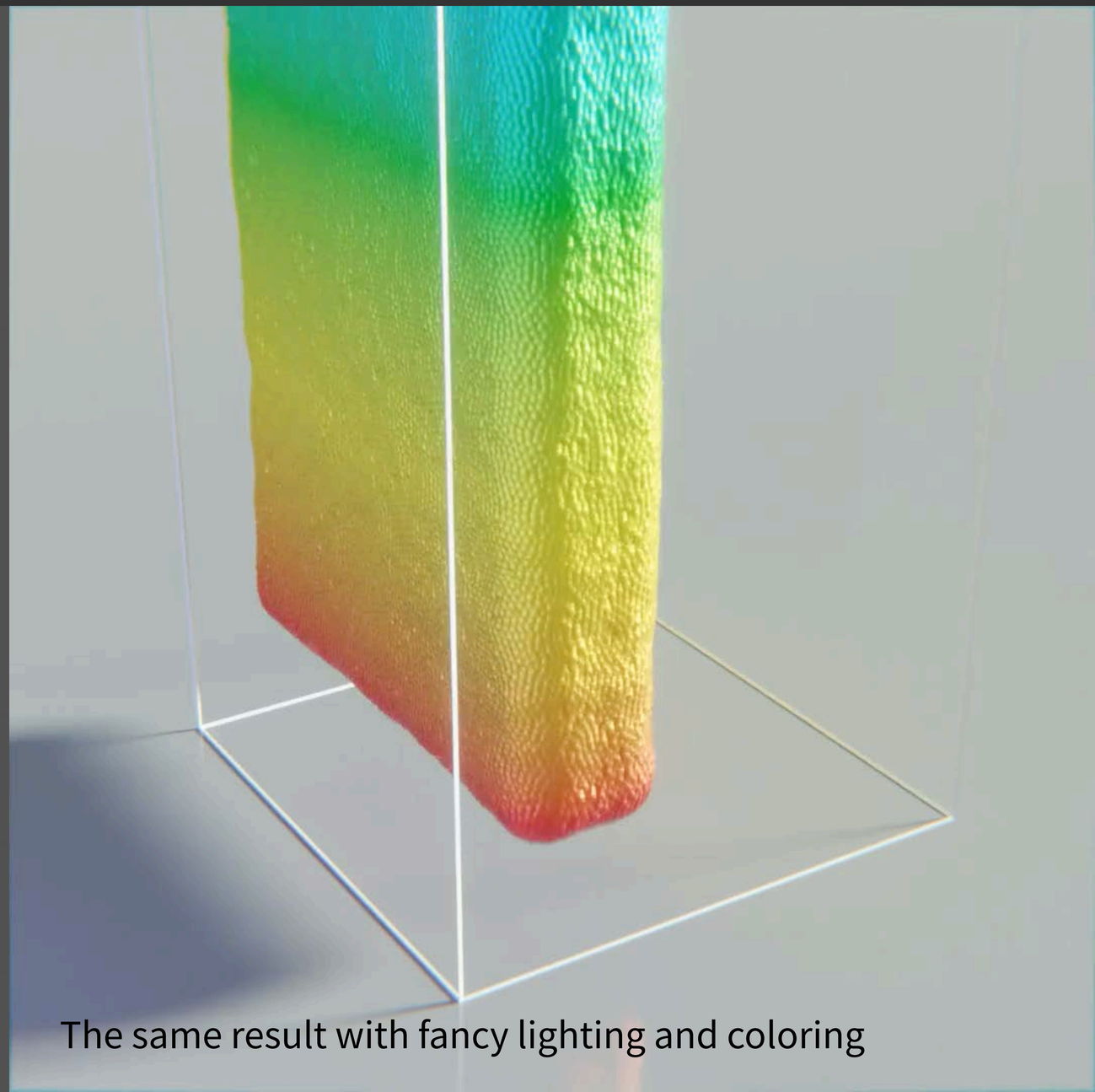
- Make presentations more memorable, leading to **better communication**
- Capture the audience's **attention**

https://www.youtube.com/watch?v=HfHC0wNYry8

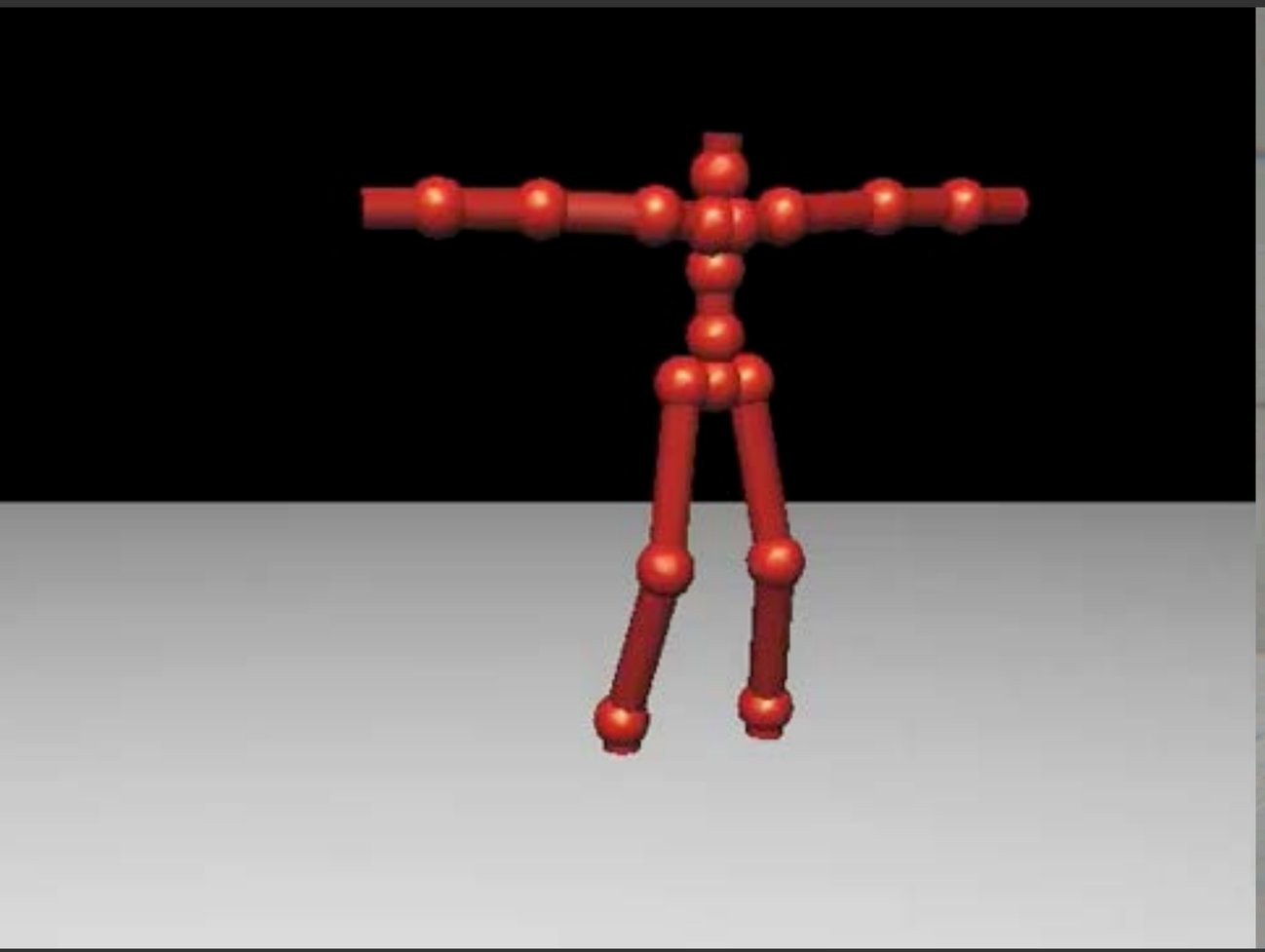
Which is better?

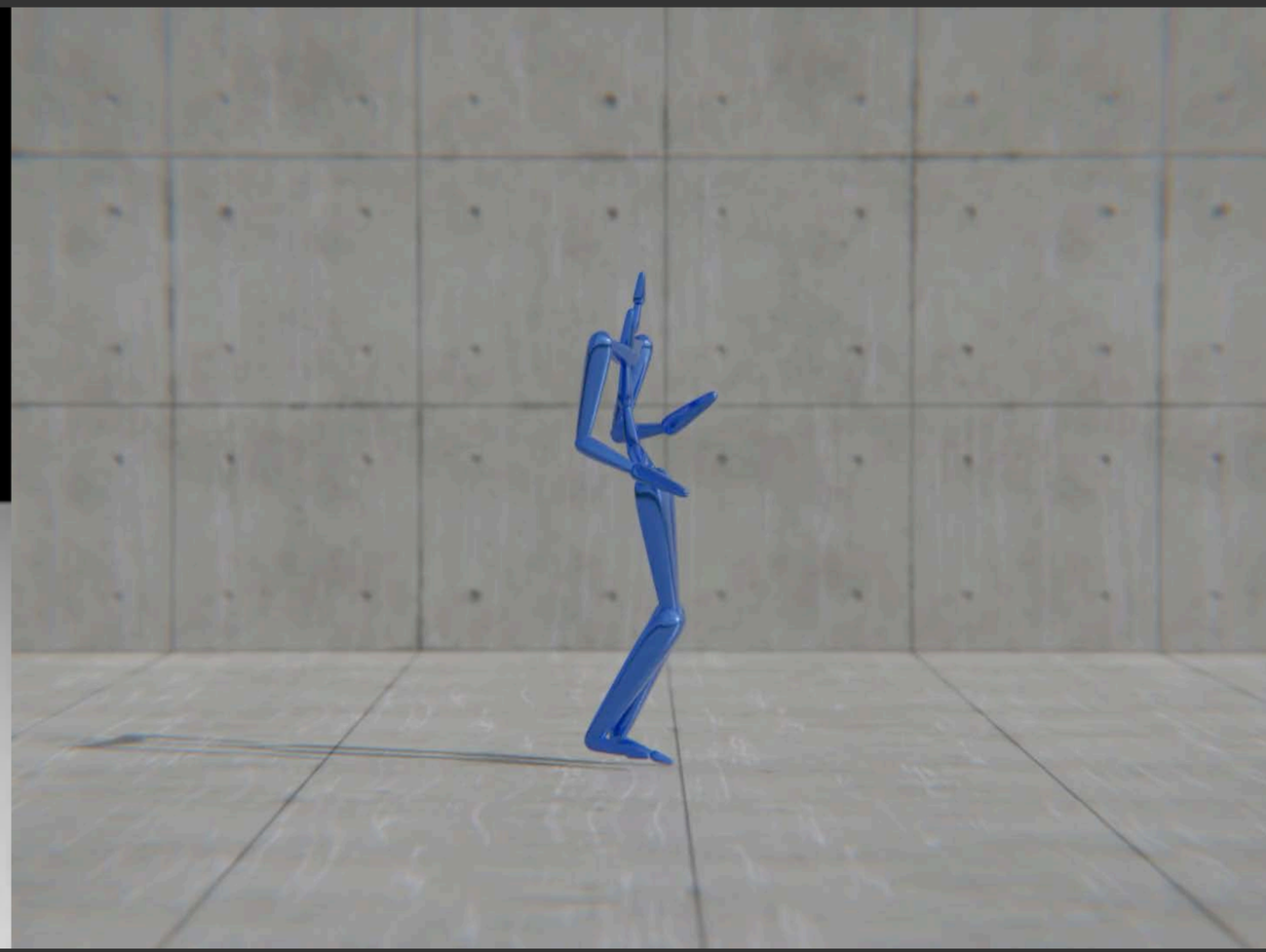


Particle-based fluid simulation result, showing each particle as a black dot



The same result with fancy lighting and coloring





Which do you want to retweet?

Scenarios: When to Use 3DCG?

Illustrating Complex Concepts

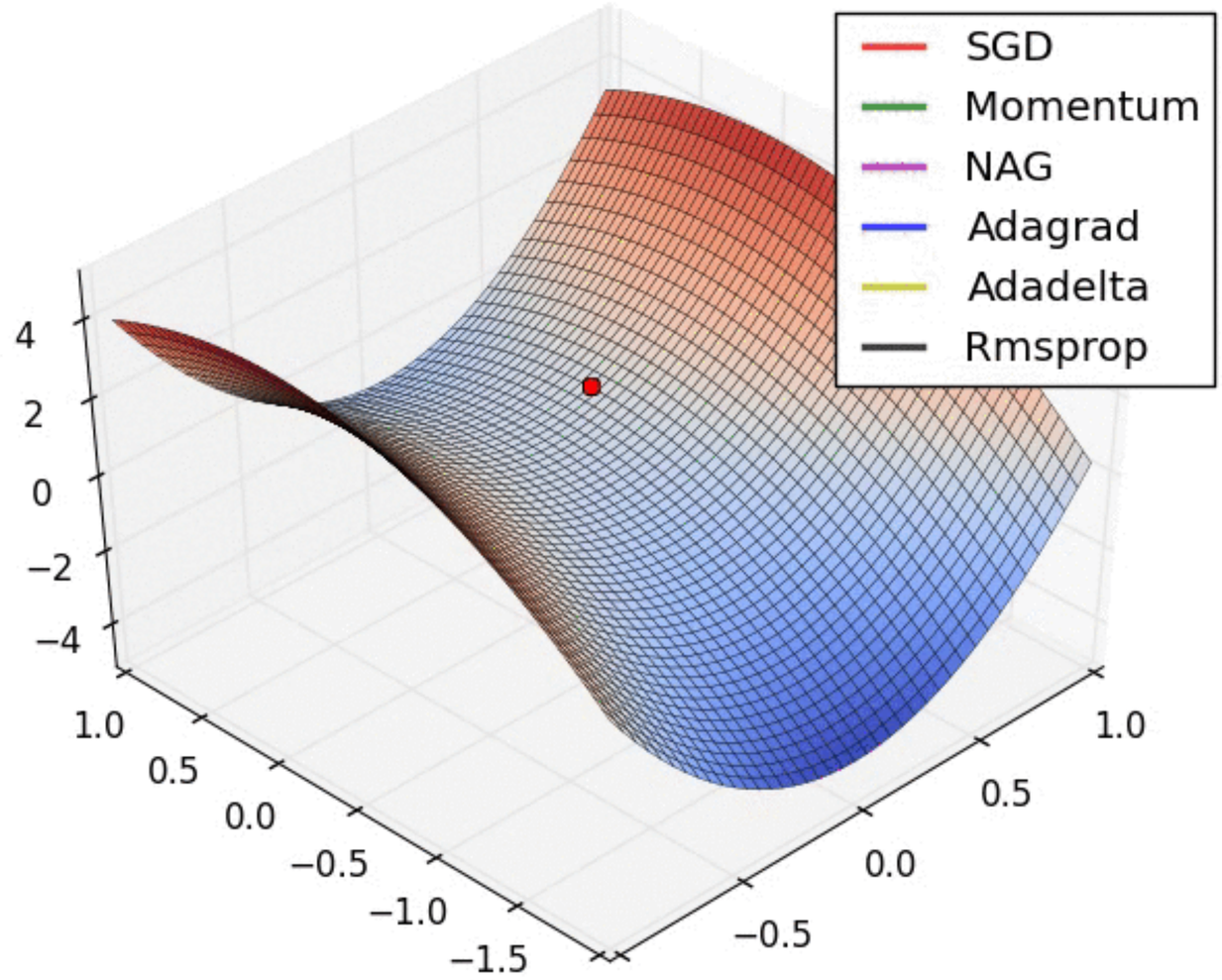
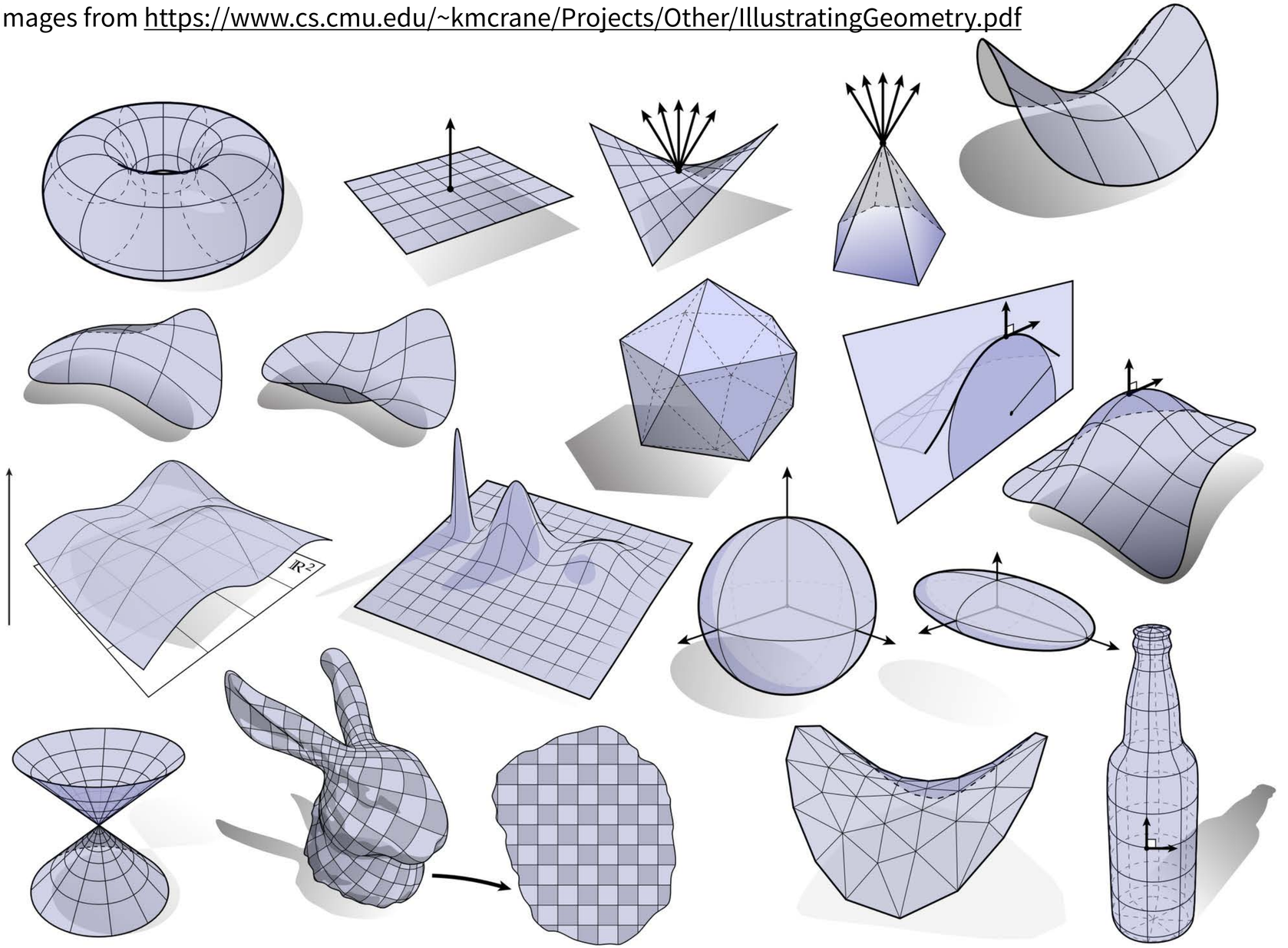


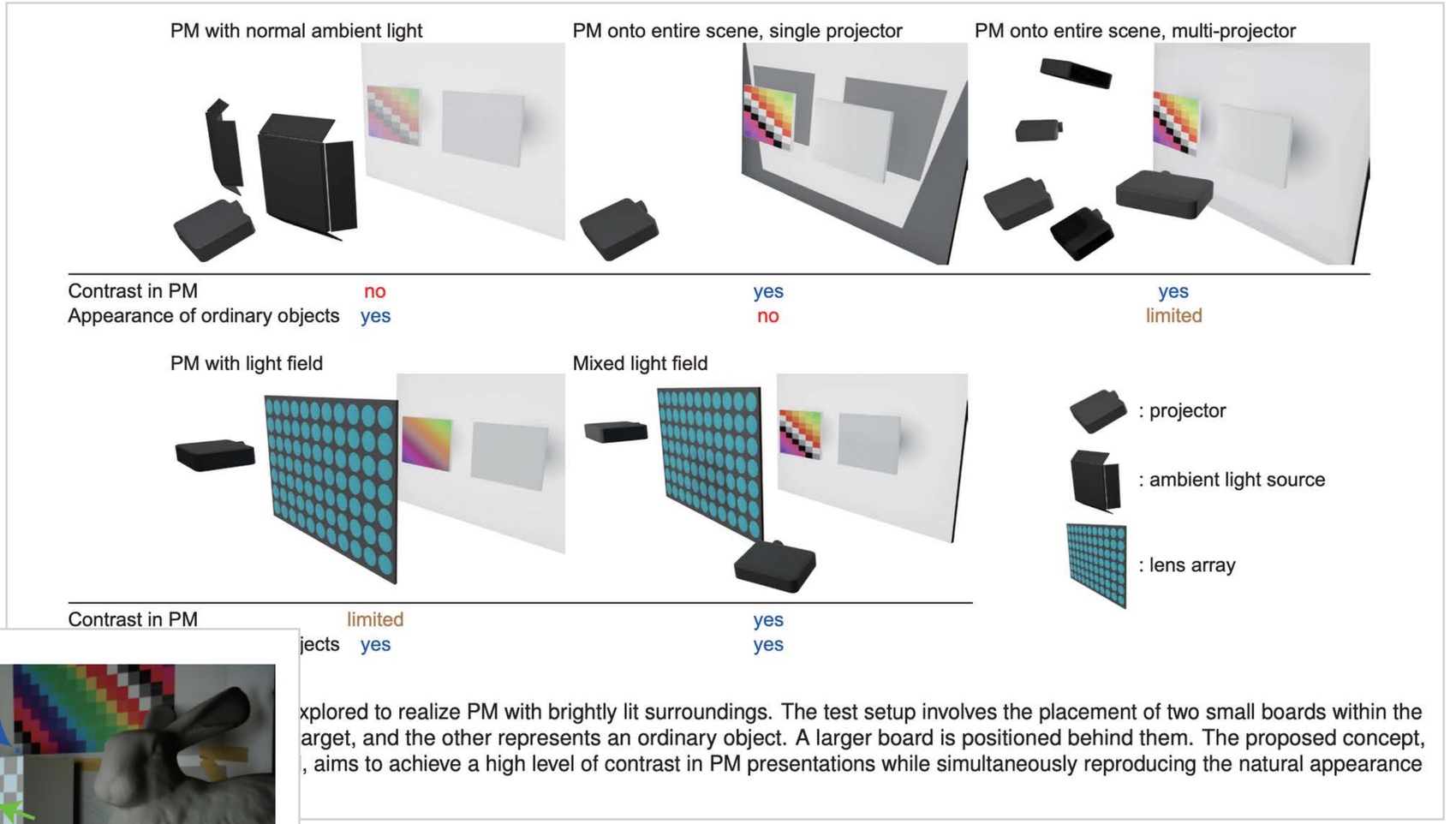
Image from https://www.ruder.io/optimizing-gradient-descent/

Illustrating Complex Concepts

Images from https://www.cs.cmu.edu/~kmcrane/Projects/Other/IllustratingGeometry.pdf



Illustrating Complex Experimental Settings



xplored to realize PM with brightly lit surroundings. The test setup involves the placement of two small boards within the arget, and the other represents an ordinary object. A larger board is positioned behind them. The proposed concept, , aims to achieve a high level of contrast in PM presentations while simultaneously reproducing the natural appearance

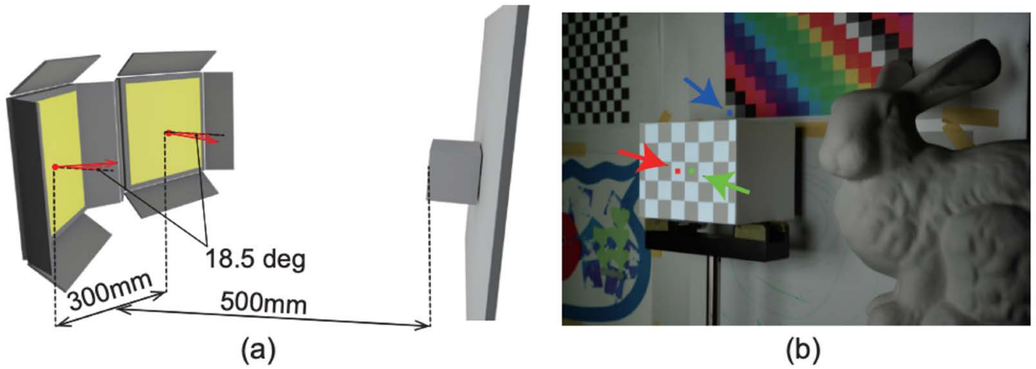
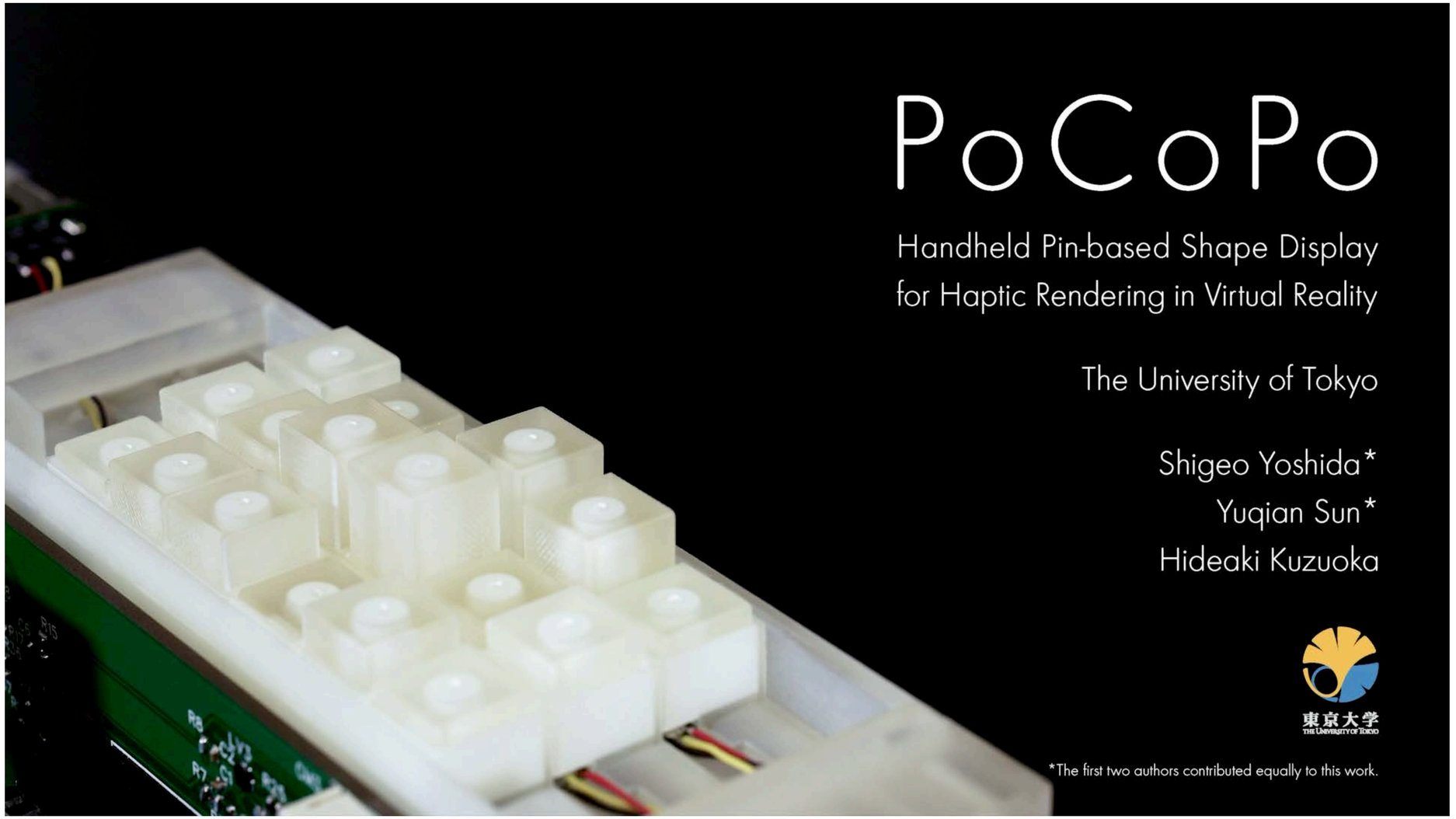


Fig. 12: (a) Arrangement of panel lights in the scene. (b) Measured point for the irradiance and contrast evaluations. *White*, *black*, and *background* values were measured at the red, green, and blue points, respectively.

Images from https://ieeexplore.ieee.org/document/10461983

Mechanical Assemblies: Step-by-Step Assemble



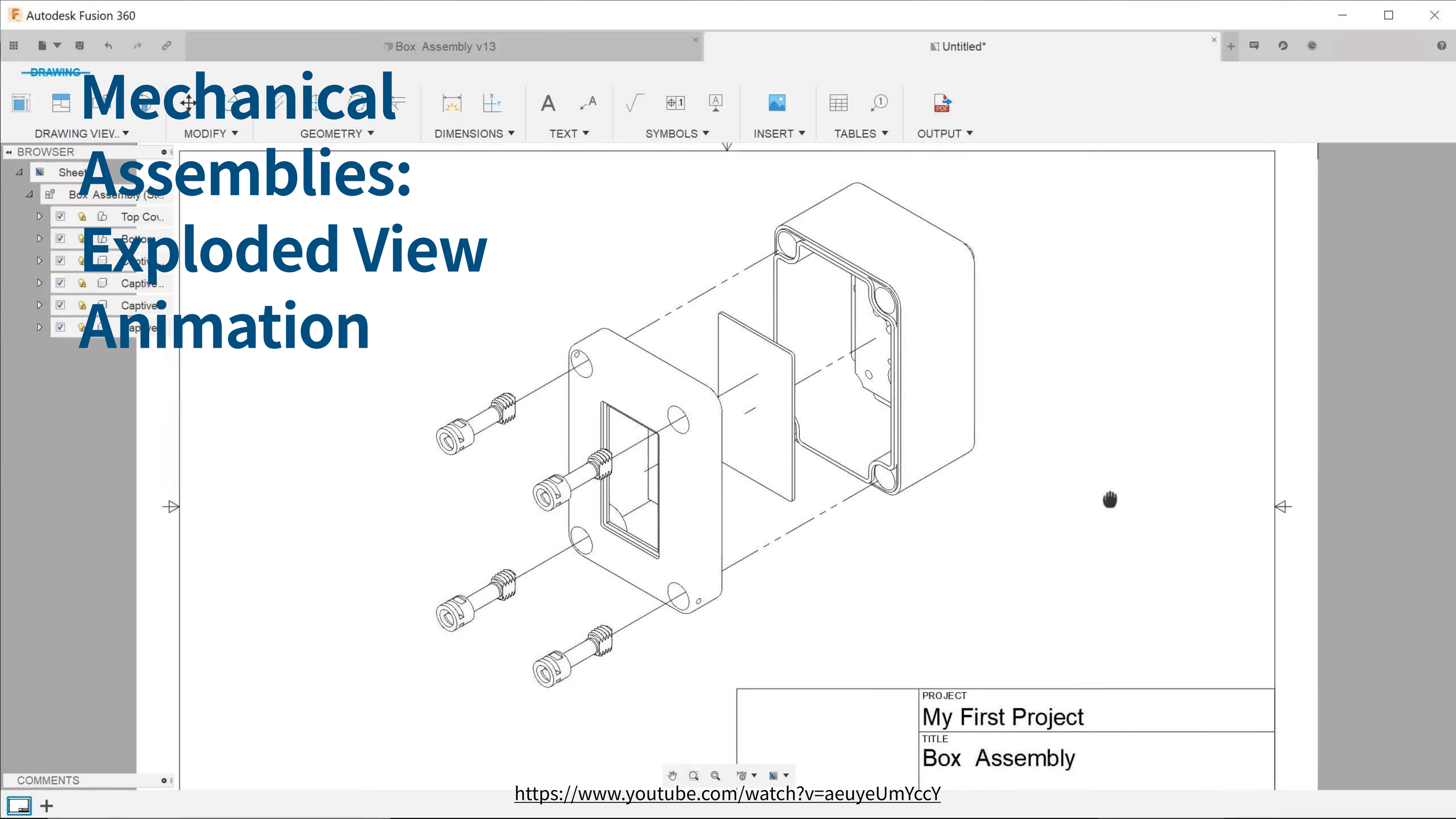
1:01—1:34 https://www.youtube.com/watch?v=Cd09lh2cXnc

Mechanical Assemblies: Exploded View Animation

- Help understanding of the assembly, and
- Simply attractive

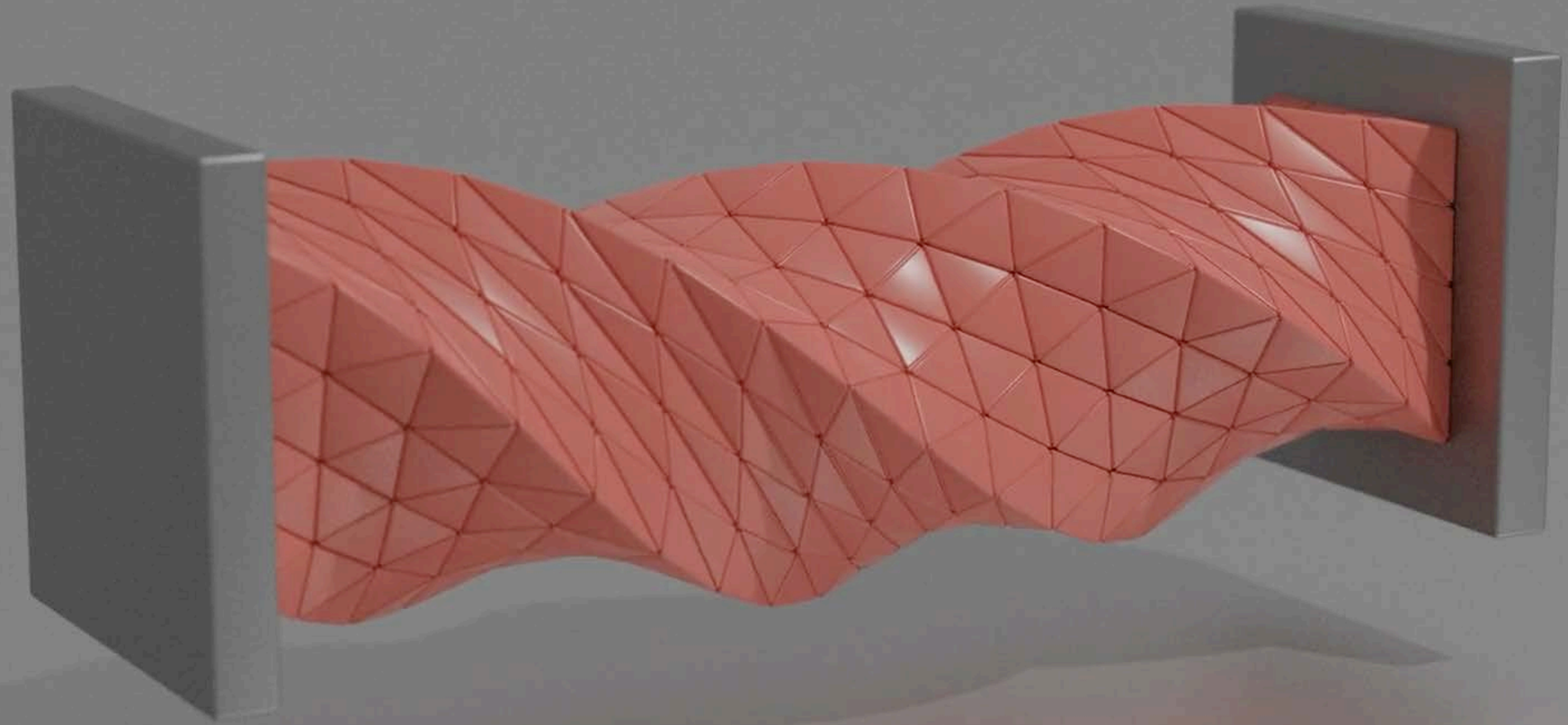
The 3D model was provided by George Kayesi for tutorial purposes
https://www.youtube.com/watch?v=IC0-9b0Rv2g

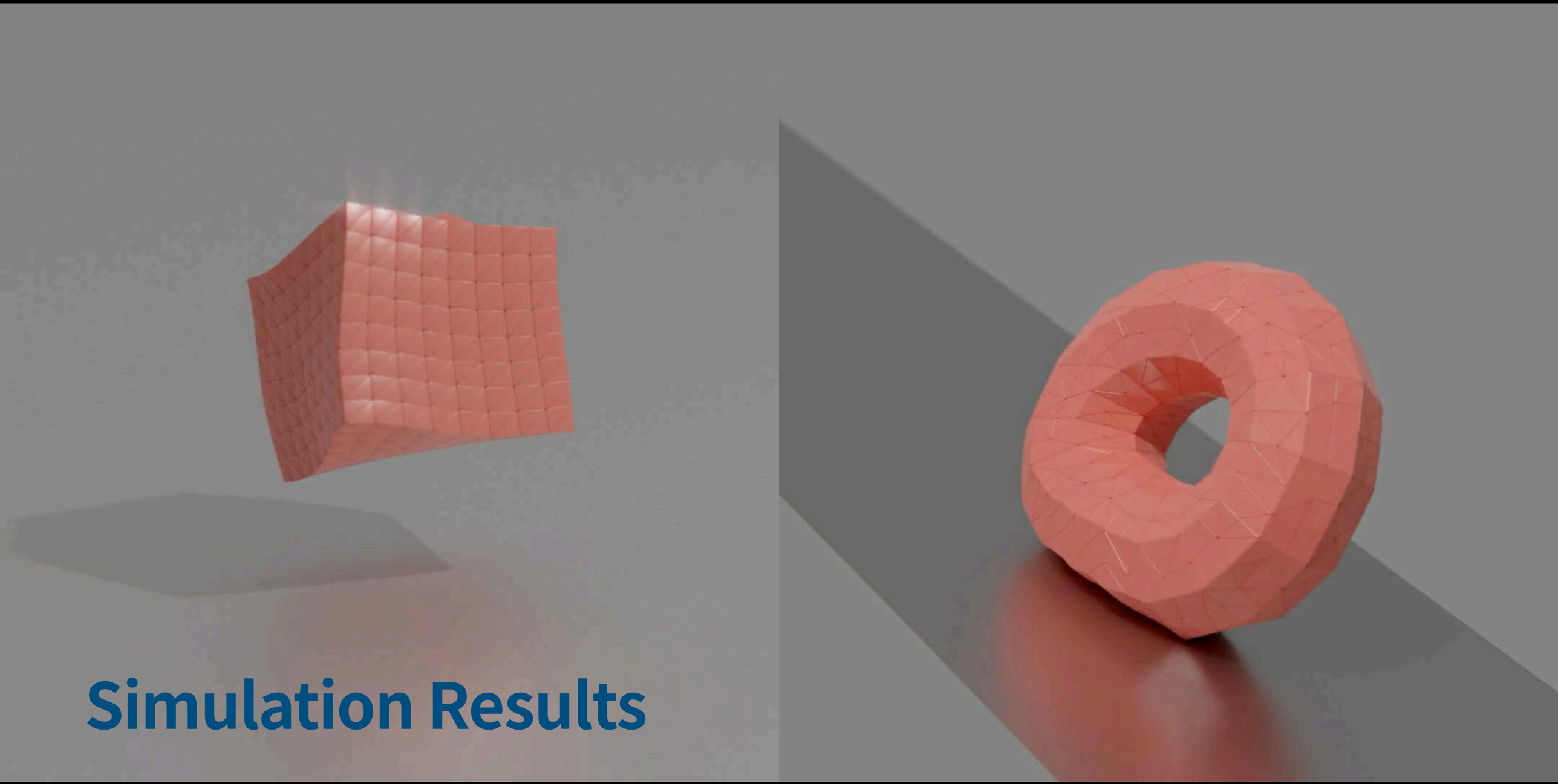
Mechanical Assemblies: Exploded View Animation



https://www.youtube.com/watch?v=aeuyeUmYccY

Simulation Results

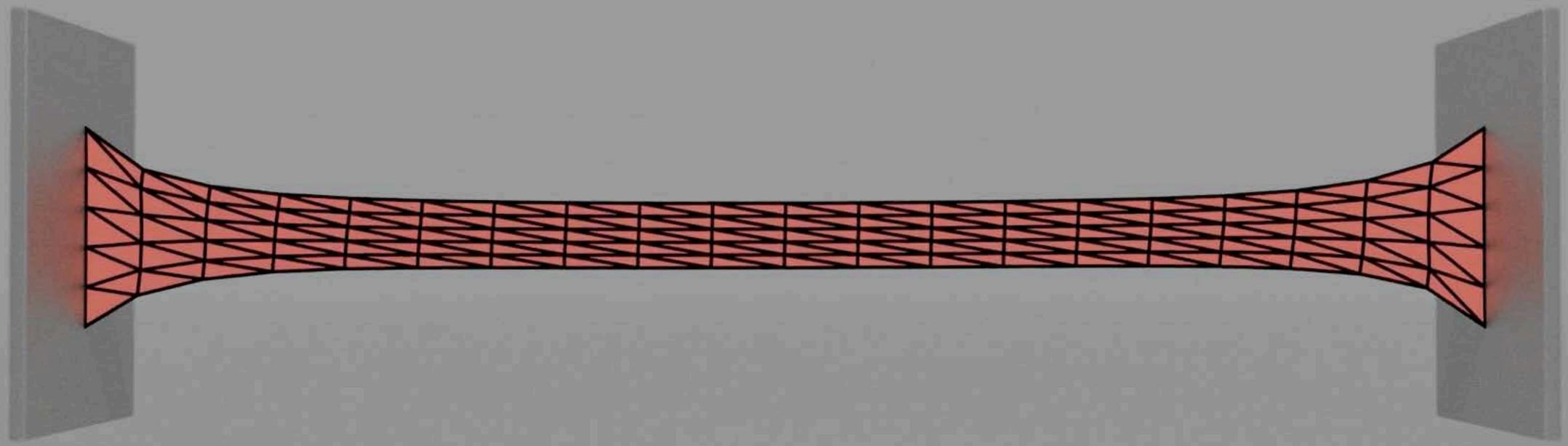




Simulation Results

Simulation Results

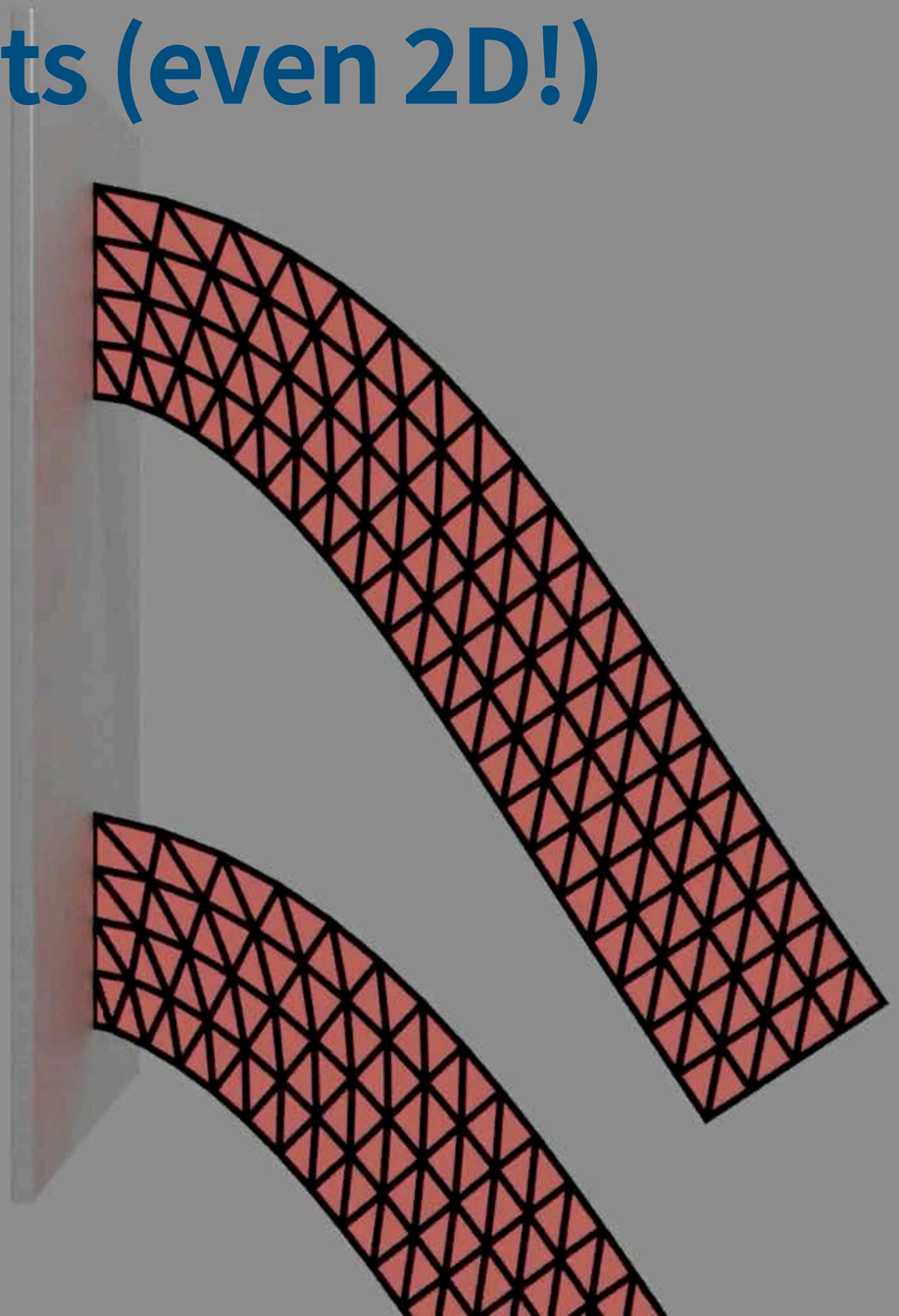
Simulation Results (even 2D!)



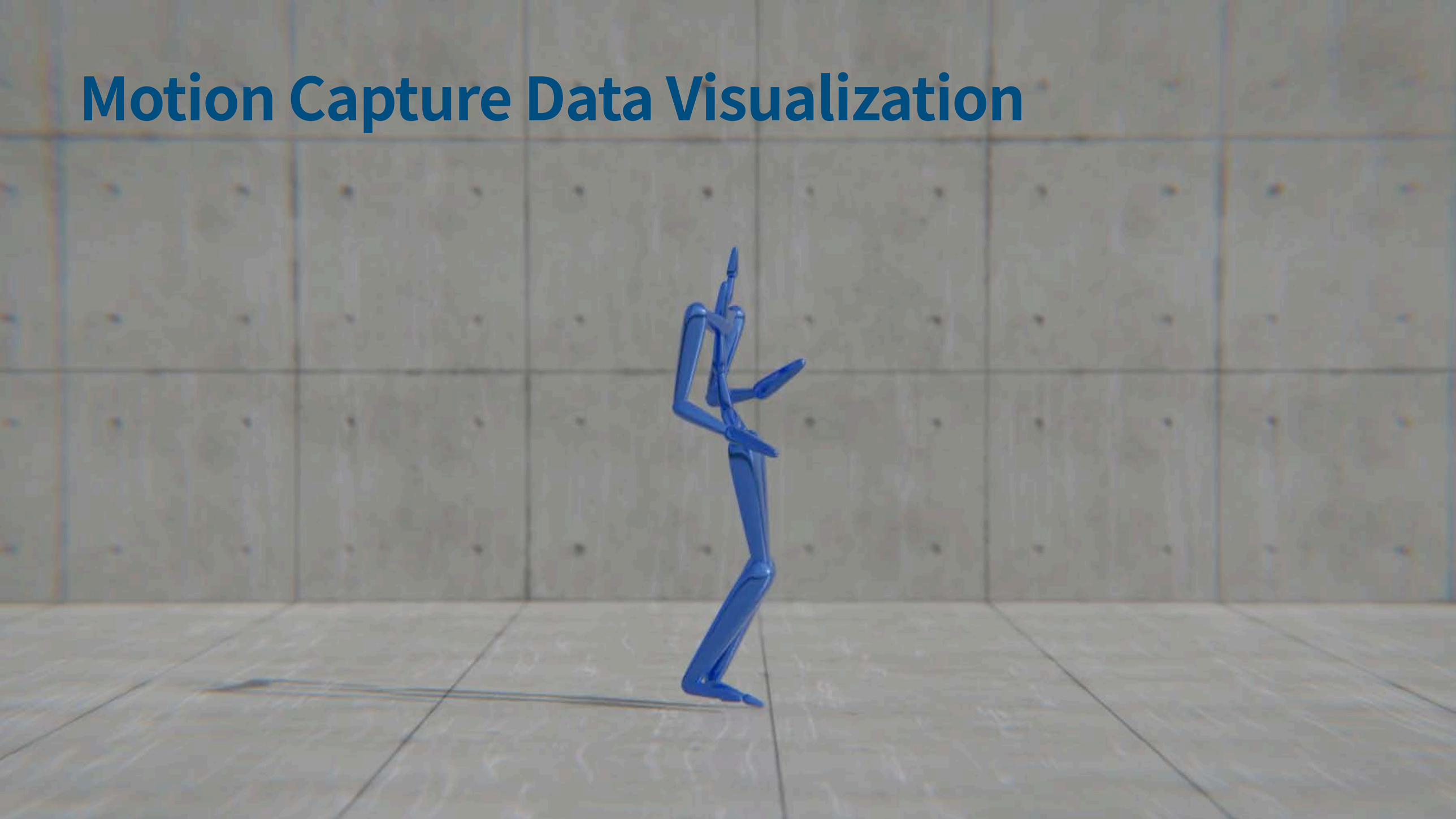
Simulation Results (even 2D!)

Co-Rotational

St. Venant-Kirchhoff



Motion Capture Data Visualization



3DCG Software

Blender

- Comprehensive 3DCG software
- **Free** and open source (GPL)
- Programmer-friendly (next slide)

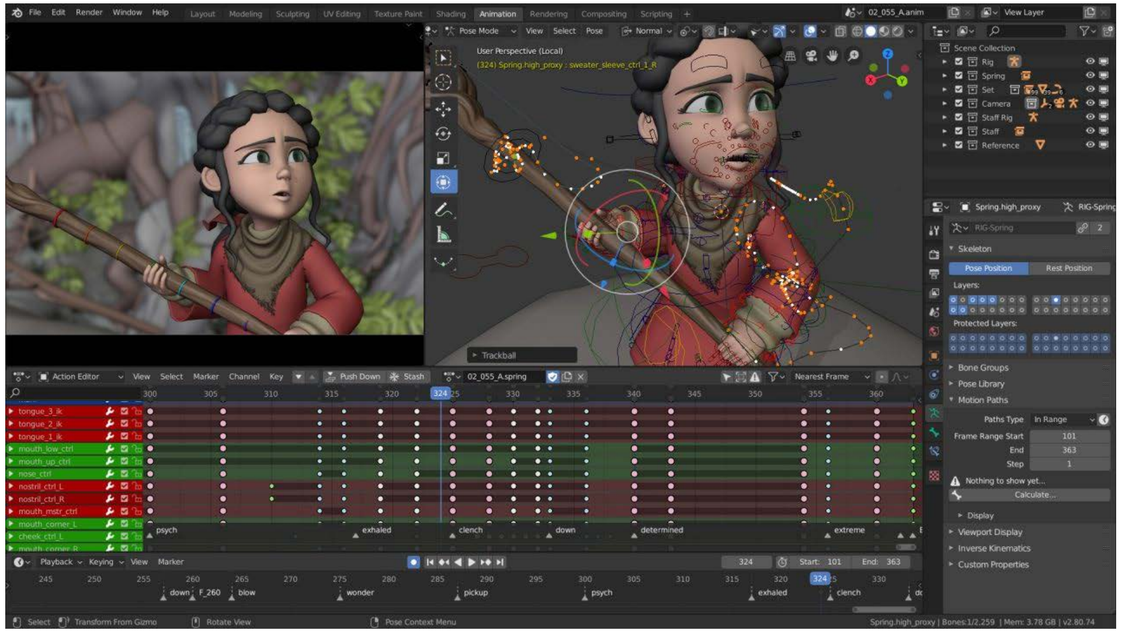


Image from https://www.blender.org/

Blender is … Programmer-Friendly! 🤝

- **Easy to install using command line (see the right)**
- Able to run on (display-less) computing servers (e.g., ABCI, Miyabi, AWS, etc.)
- Everything can be manipulated and automated via **Python APIs**
- **Programmer-friendly also means AI-friendly!!**

Other Software (Not Free)

- **3DCG animation and VFX**
  - Maya
  - 3ds Max
  - Houdini
  - …
- **Game engine**
  - Unreal Engine
  - Unity
  - ...
- **CAD**
  - Autodesk Fusion
  - Rhinoceros
  - AutoCAD
  - Solidworks
  - …

3DCG Tips for Researchers

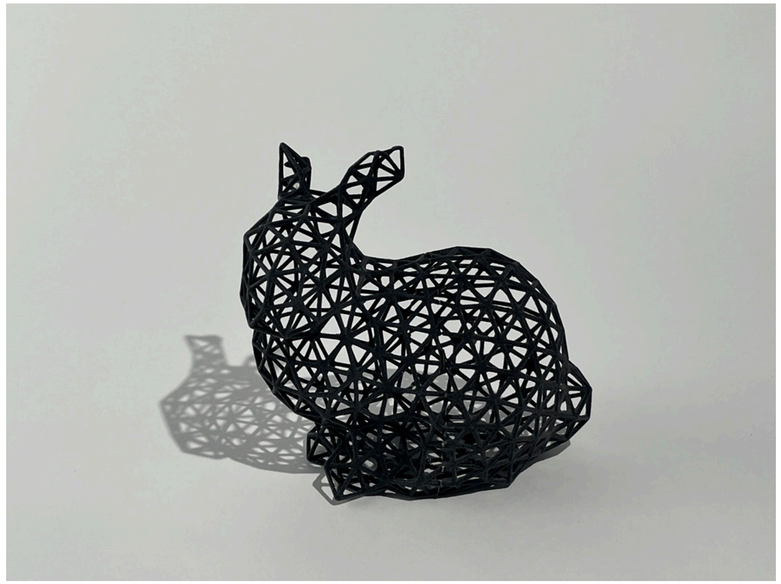
How to Create Good Illustrations?

Lighting Tips | Detailed in the following slides

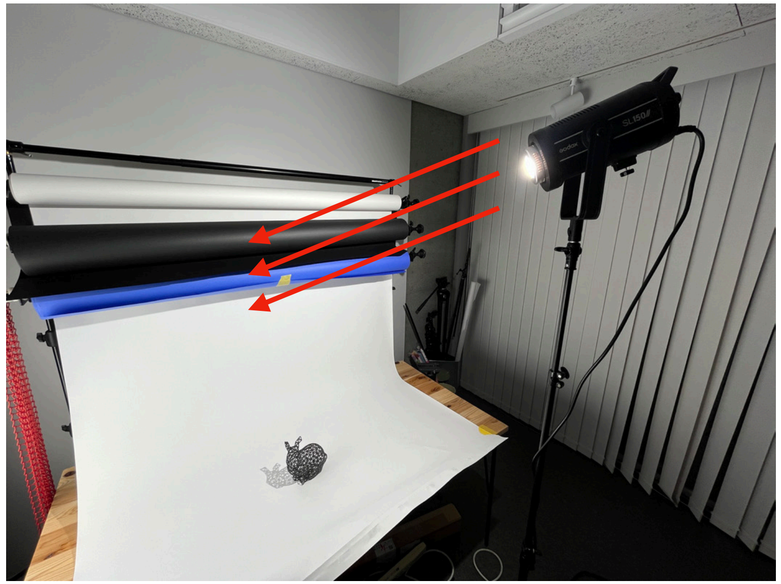
- Use soft shadow (avoid hard shadow)
- Use cyclorama (infinity curve) background
- Environmental light is easy and useful

Do you remember? — Week5 [Figures] by Koya Narumi

Avoid a non-diffused light



Strong shadow → Bad

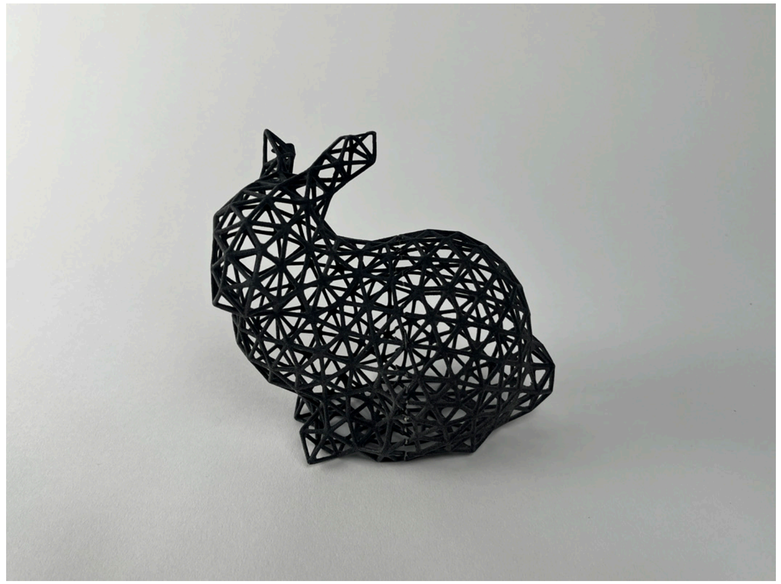


Non-diffused light → Bad

Don’t use a strong, non-diffused light like a summer beach.

Do you remember? — Week5 [Figures] by Koya Narumi

Use a diffused light



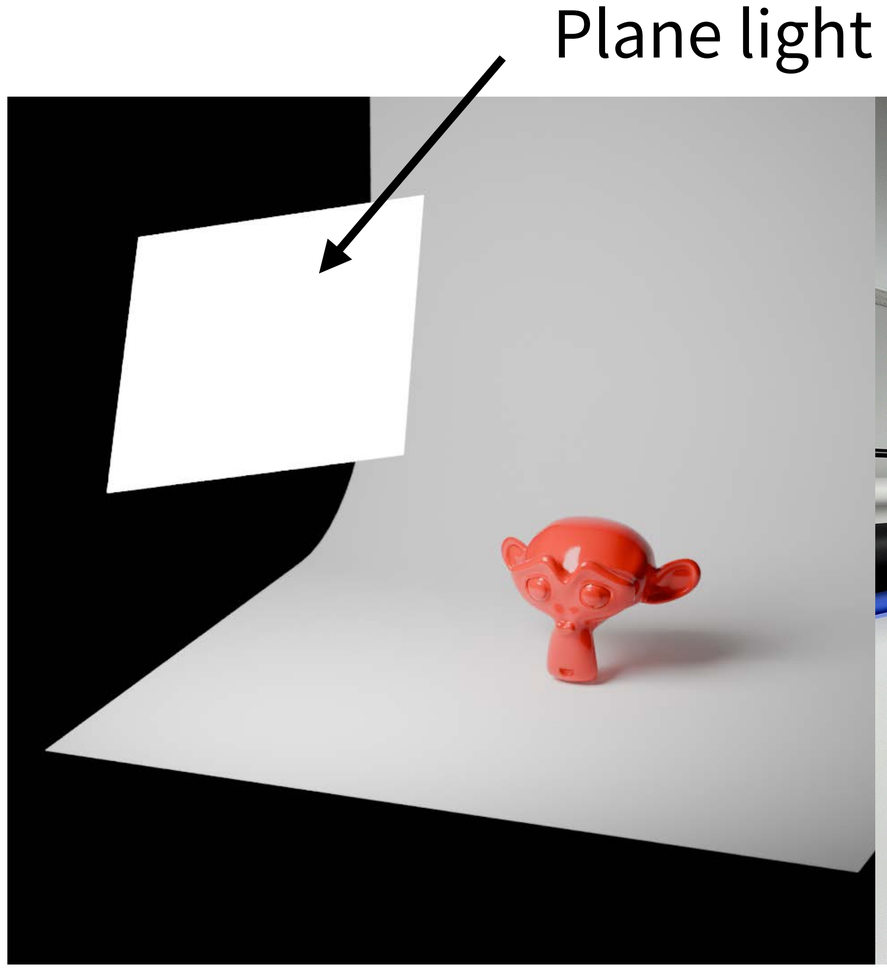
Soft shadow → Good

Diffused light → Good

The diffused light like a cloudy sky **is much better.**

Lighting: Soft Shadow

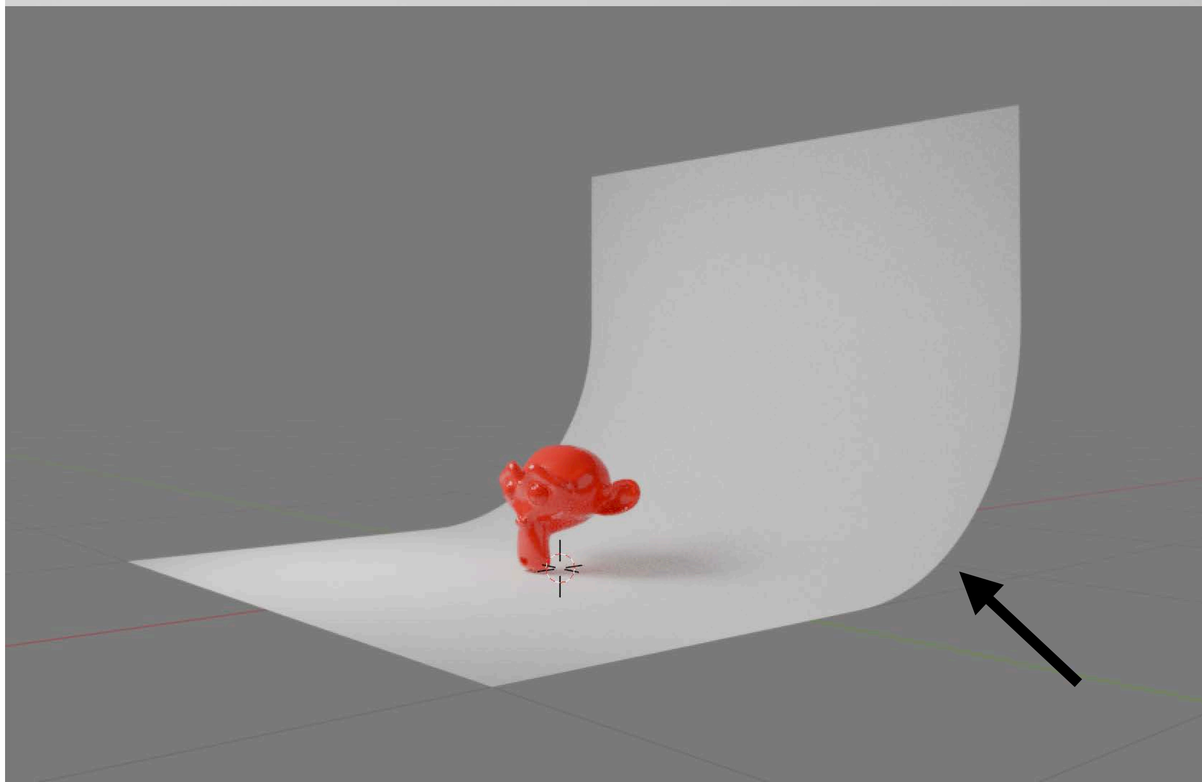
- 3DCG can easily create soft shadows by using **plane lights**
- No need of expensive equipments as in real-world photography🥰



No need to buy this

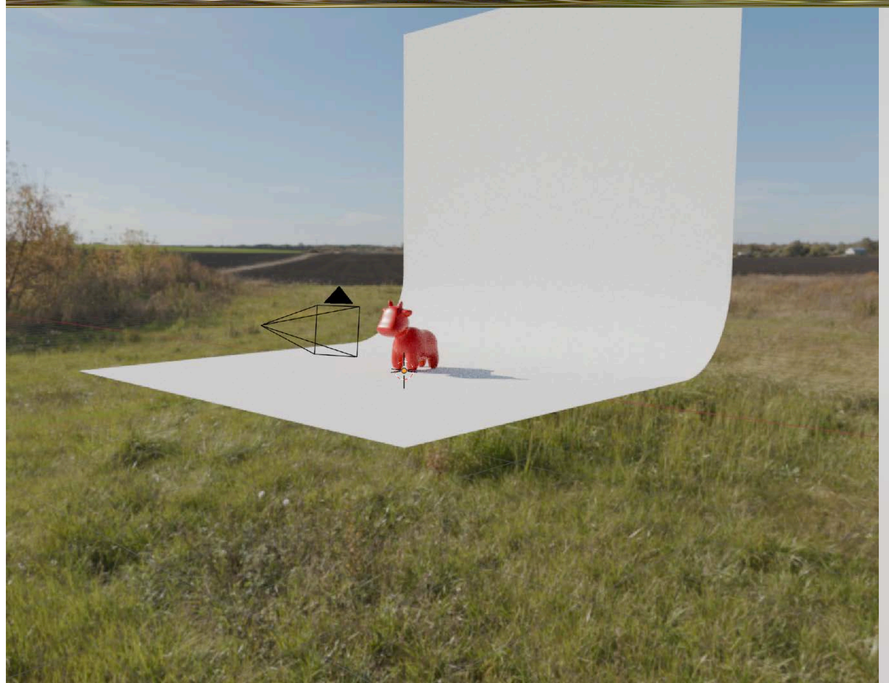
Lighting: Cyclorama (Infinity Curve) Background

- Curved background to create **the illusion of an infinite space**
- Used in photography
- Also useful in 3DCG

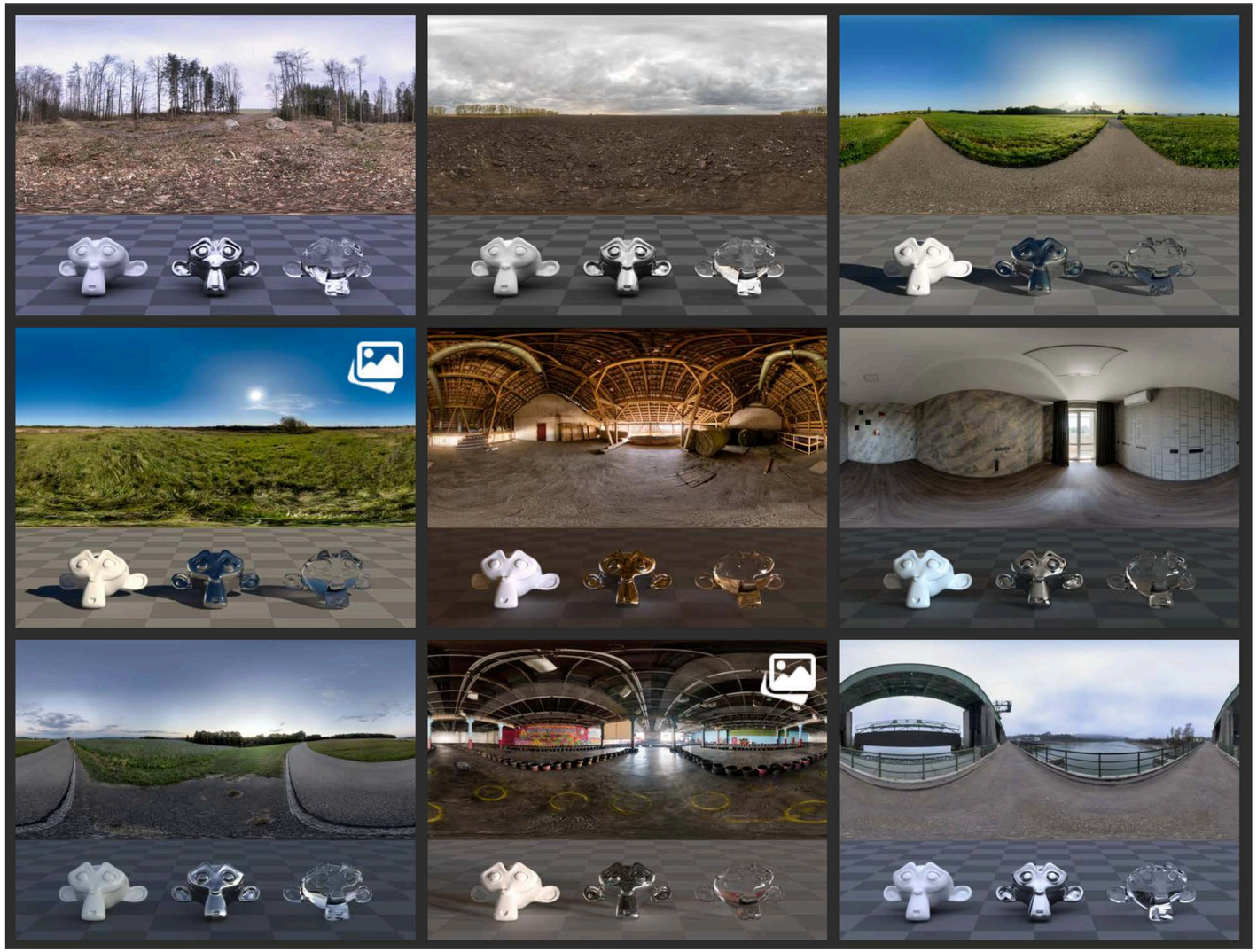


Lighting: Environmental Map

- … is a texture that represents the surrounding environment of a 3D scene
- Highly useful for creating realistic lighting **(very easy🥹)**



Lighting: Where to Find HDR Images?



https://hdri-haven.com/

Lighting: File Format for Environmental Maps

- **High Dynamic Range (HDR) image format (.hdr)** is used

- **Typical formats: unsigned int (8 bit/channel)** — $[0,255]$
  - 😢 Cannot represent very bright (> 255) light sources (overexposure)
- **HDR format: float (32 bit/channel)** — $[0,\infty)$
  - 😀 Can represent more-than-1.0 values! 💪

Materials Tips

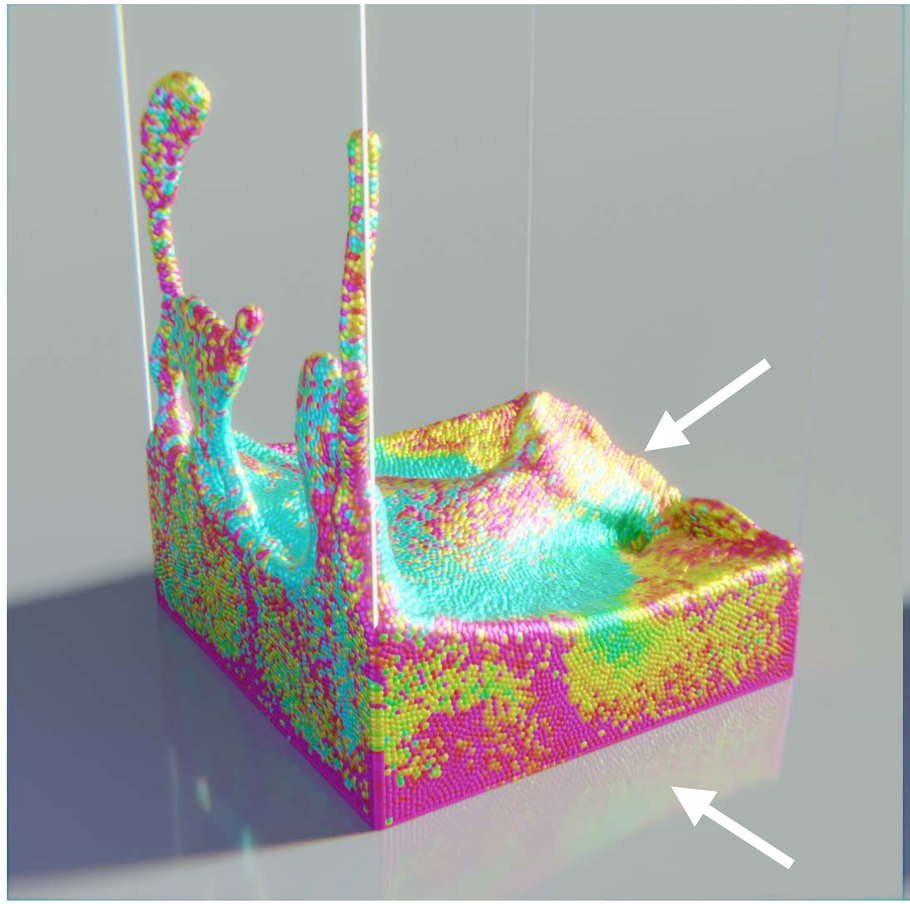
- **Reflection** helps better shape perception 💡
- But be careful ⚠ Reflection increases **visual clutter**

with reflection

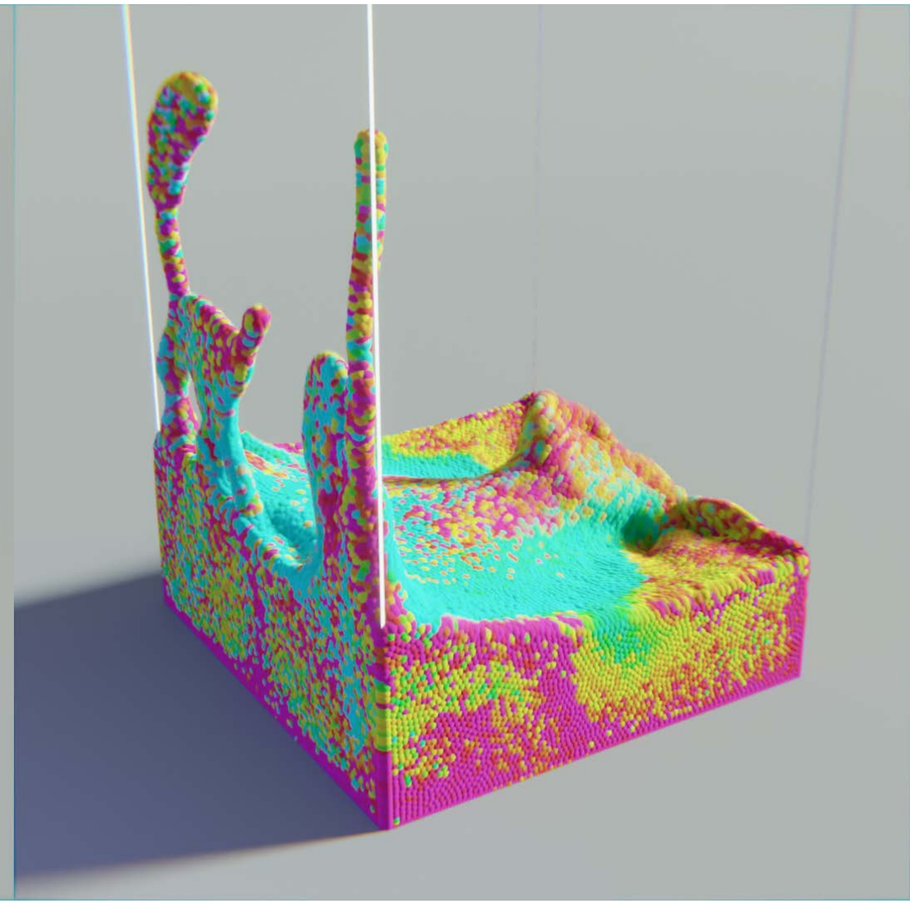
without reflection

Materials Tips

- **Reflection** helps better shape perception 💡
- But be careful ⚠ Reflection increases **visual clutter**



with reflection

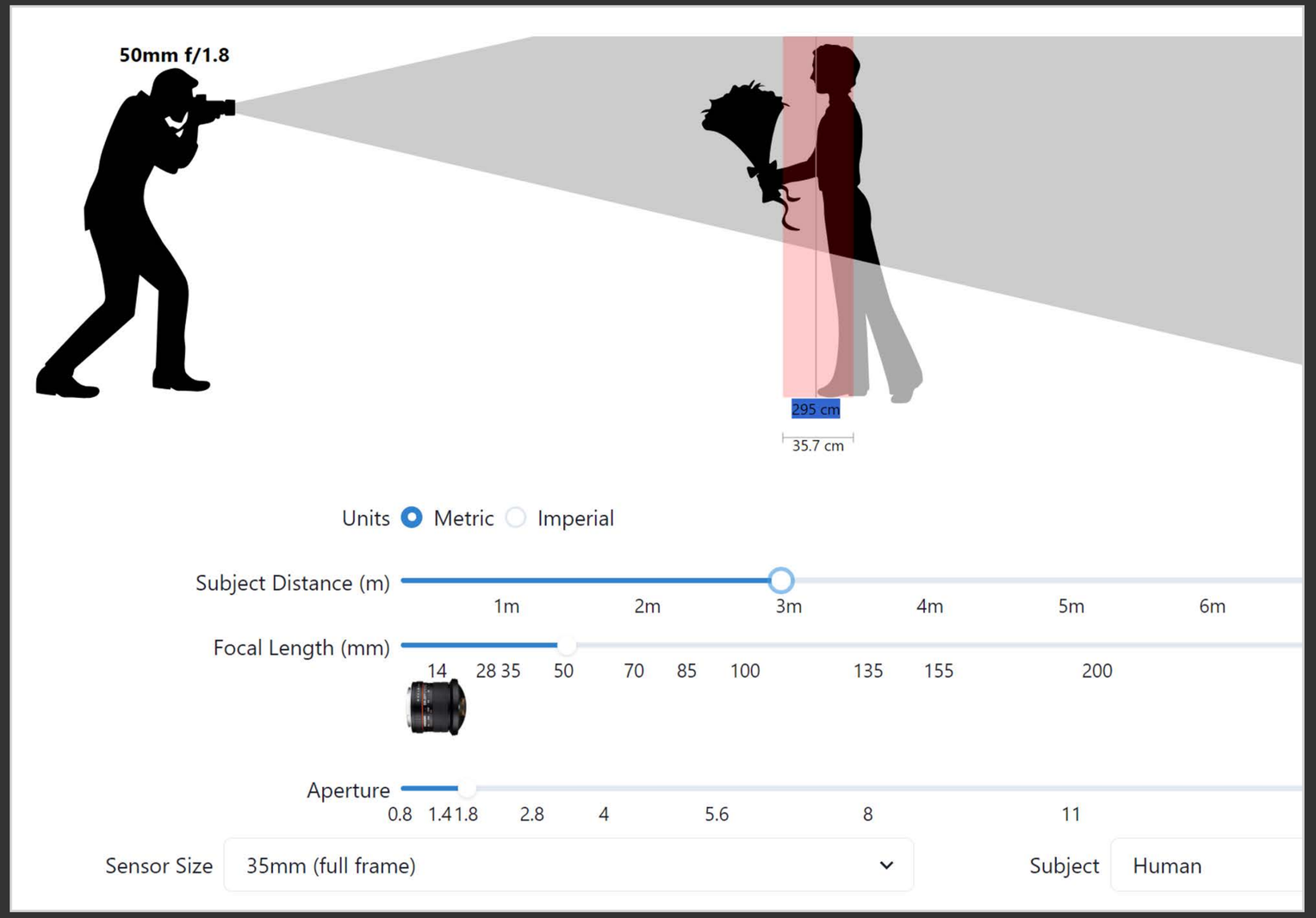


without reflection

Camera Tips: Depth Blur

- Use **depth blur** for …
  - Guiding viewers' attention 💡
  - Creating emotional expressions 🥹
- But avoid using depth blur in most technical illustrations ⚠️

Depth of Field (in real world)



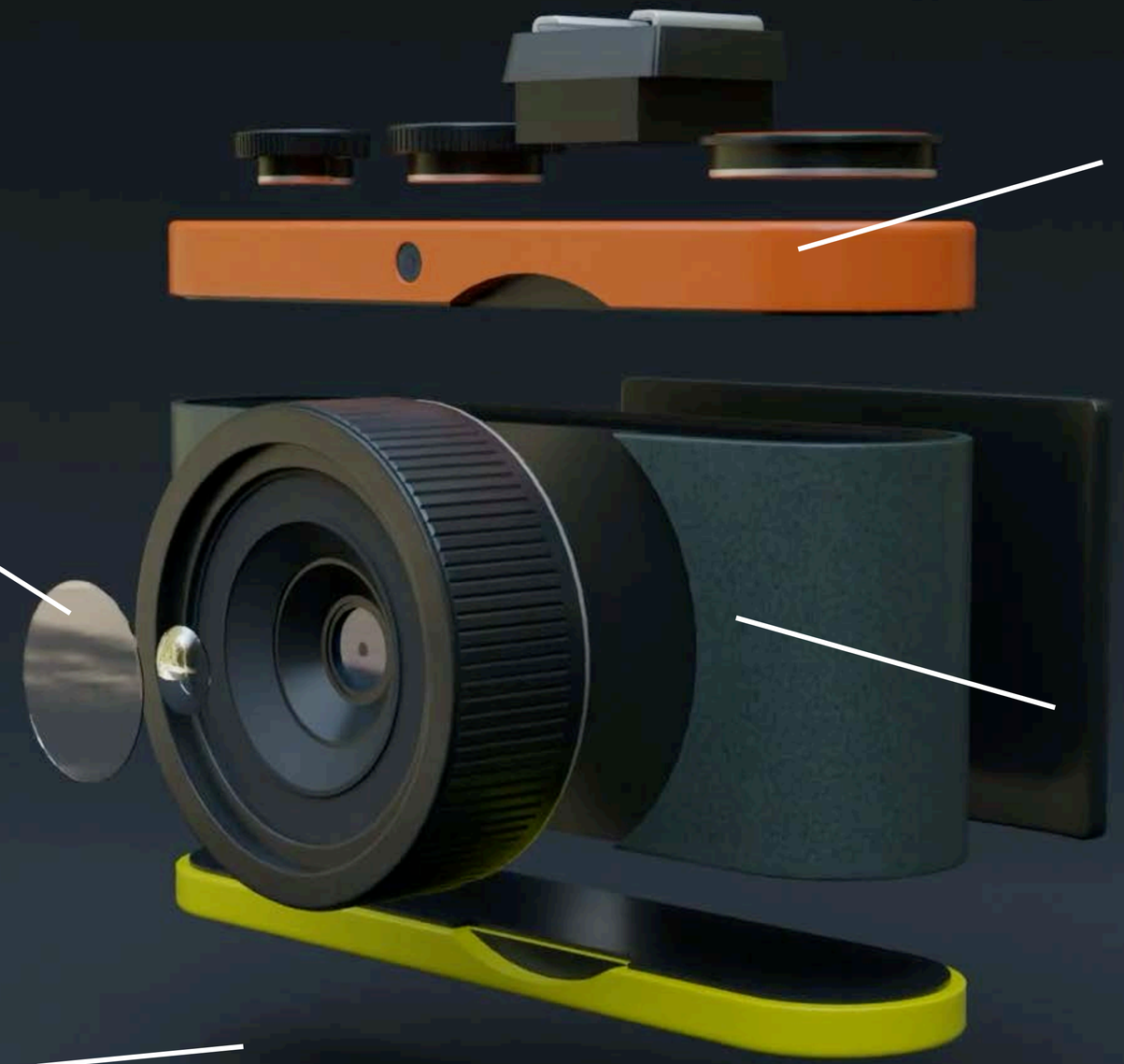
https://jherr.github.io/depth-of-field/

Scripting Tips

- **Everything that you can do with GUI can be done via Python API**
  - Automation is easy
  - Once scripting is done, it runs even on computing servers (e.g., ABCI)
- **It is easy to integrate Blender into your data processing pipeline**
  - E.g., Checkpoint visualization during the training of 3D-related ML models (3D shape generation, point cloud segmentation, human motion generation, etc.)
  - E.g., Visualization of fluid simulation results from various viewpoints (which is useful to check and validate the simulation behavior)

Demonstration

The 3D model was provided by George Kayesi for tutorial purposes
https://www.youtube.com/watch?v=IC0-9b0Rv2g



Exploded view animation

(1) helps understanding of the assembly, and

(2) is attractive

Environmental map is used to create fancy reflection

Slight rotation helps shape perception

Cyclorama (infinity curve) background

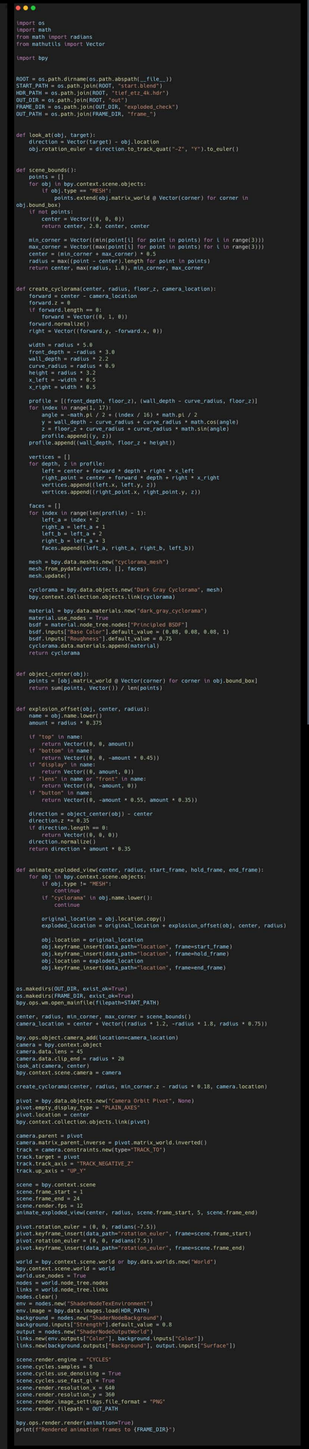
Soft shadow is often preferred

Live Programming by Codex

Knowing 3DCG tips helps you know what to ask AI

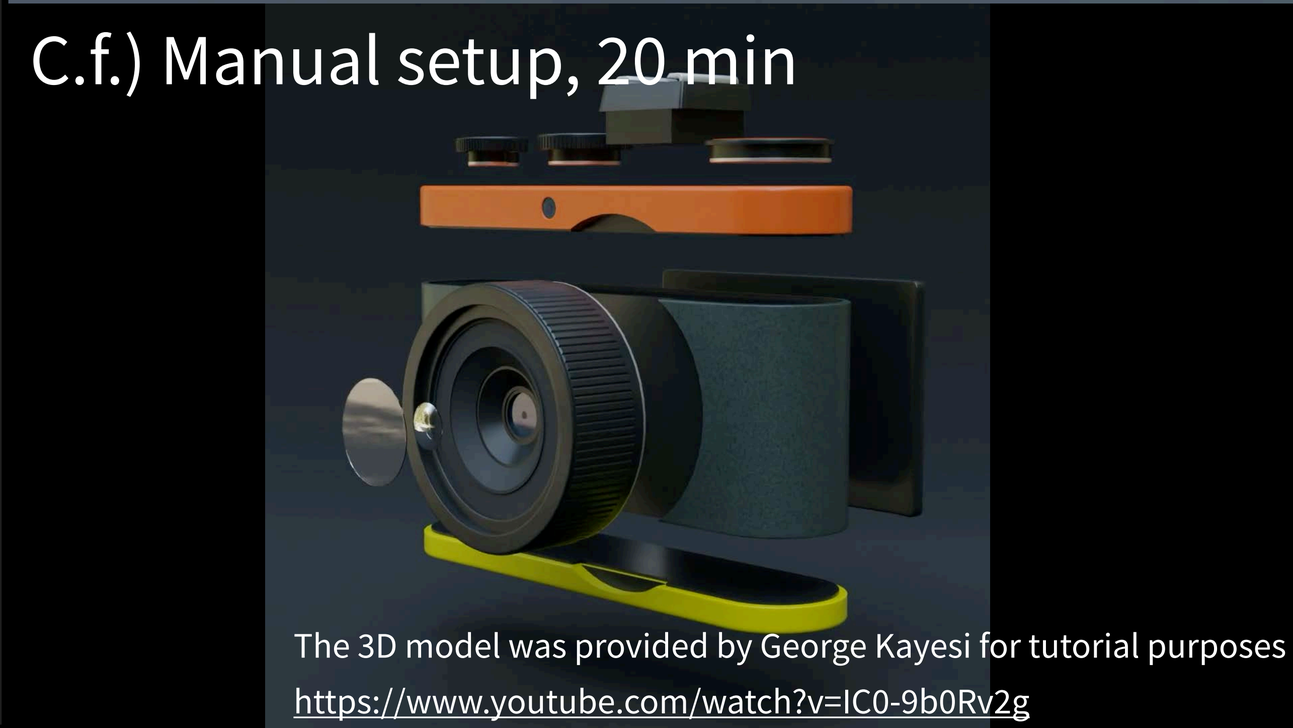
Codex wrote 200 lines

```python
import os
import math
from math import radians
from mathutils import Vector

import bpy


ROOT = os.path.dirname(os.path.abspath(__file__))
START_PATH = os.path.join(ROOT, "start.blend")
HDR_PATH = os.path.join(ROOT, "tief_etz_4k.hdr")
OUT_DIR = os.path.join(ROOT, "out")
FRAME_DIR = os.path.join(OUT_DIR, "exploded_check")
OUT_PATH = os.path.join(FRAME_DIR, "frame_")


def look_at(obj, target):
    direction = Vector(target) - obj.location
    obj.rotation_euler = direction.to_track_quat("-Z", "Y").to_euler()


def scene_bounds():
    points = []
    for obj in bpy.context.scene.objects:
        if obj.type == "MESH":
            points.extend(obj.matrix_world @ Vector(corner) for corner in
obj.bound_box)
    if not points:
        center = Vector((0, 0, 0))
        return center, 2.0, center, center

    min_corner = Vector((min(point[i] for point in points) for i in range(3)))
    max_corner = Vector((max(point[i] for point in points) for i in range(3)))
    center = (min_corner + max_corner) * 0.5
    radius = max((point - center).length for point in points)
    return center, max(radius, 1.0), min_corner, max_corner


def create_cyclorama(center, radius, floor_z, camera_location):
    forward = center - camera_location
    forward.z = 0
    if forward.length == 0:
        forward = Vector((0, 1, 0))
    forward.normalize()
    right = Vector((forward.y, -forward.x, 0))

    width = radius * 5.0
    front_depth = -radius * 3.0
    wall_depth = radius * 2.2
    curve_radius = radius * 0.9
    height = radius * 3.2
    x_left = -width * 0.5
    x_right = width * 0.5

    profile = [(front_depth, floor_z), (wall_depth - curve_radius, floor_z)]
    for index in range(1, 17):
        angle = -math.pi / 2 + (index / 16) * math.pi / 2
        y = wall_depth - curve_radius + curve_radius * math.cos(angle)
```



Coding with Codex, 20 min

C.f.) Manual setup, 20 min



The 3D model was provided by George Kayesi for tutorial purposes

https://www.youtube.com/watch?v=IC0-9b0Rv2g

Mastering 3DCG: Where to Start?

Resources

The "Donut" Tutorial

https://www.youtube.com/@blenderguru

Resources

- Exploded View Animation of Objects in Blender (Tutorial)
  - https://www.youtube.com/watch?v=IC0-9b0Rv2g
- Illustrating Geometry (Keenan Crane)
  - https://www.cs.cmu.edu/~kmcrane/Projects/Other/IllustratingGeometry.pdf
- Multi-scale Multi-physics Heart Simulator UT-Heart
  - https://www.youtube.com/watch?v=2LPboySOSvo
- Blender Python scripting examples
  - https://github.com/yuki-koyama/blender-cli-rendering

Summary

In general, good illustrations ...

1. help audience better understand your research 💡 , and
2. increase presentation appeal 💖 (→ buzz 🧨).

3DCG is (sometimes) an effective technique for these purposes.