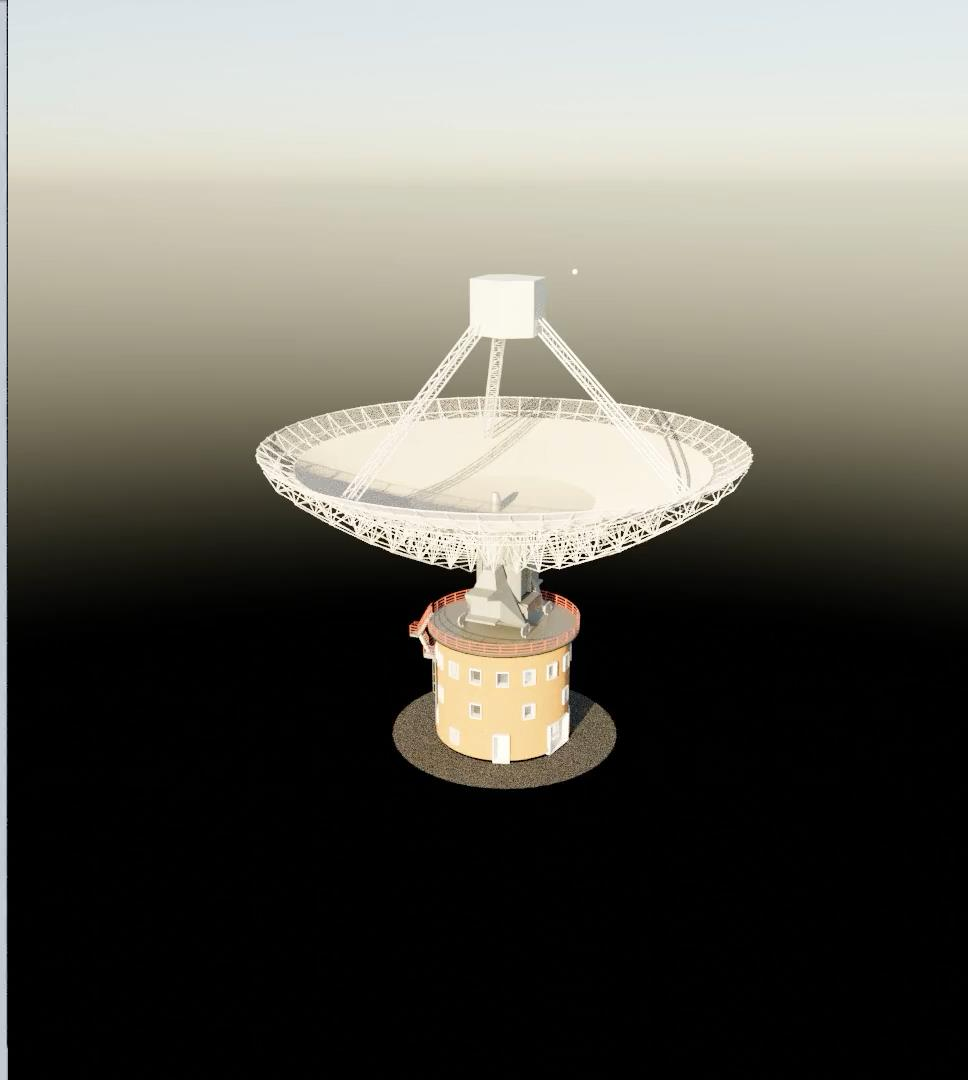
An introduction to Blender

May 2025

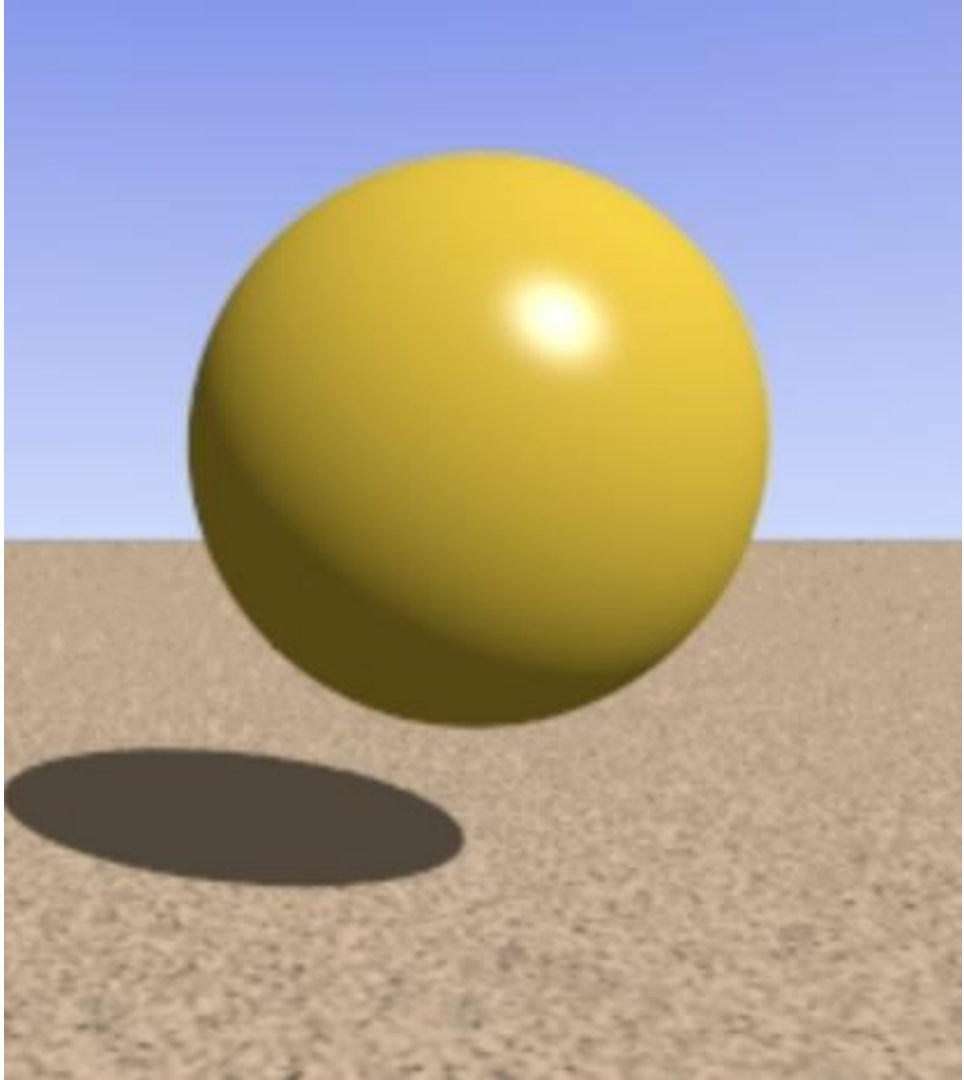
George Hobbs



Starting with POVRAY

- The Persistence of Vision Raytracer
- Started in the early 1990s
- Fully script-based
- All objects are mathematically perfect (perfect spheres, not meshes)

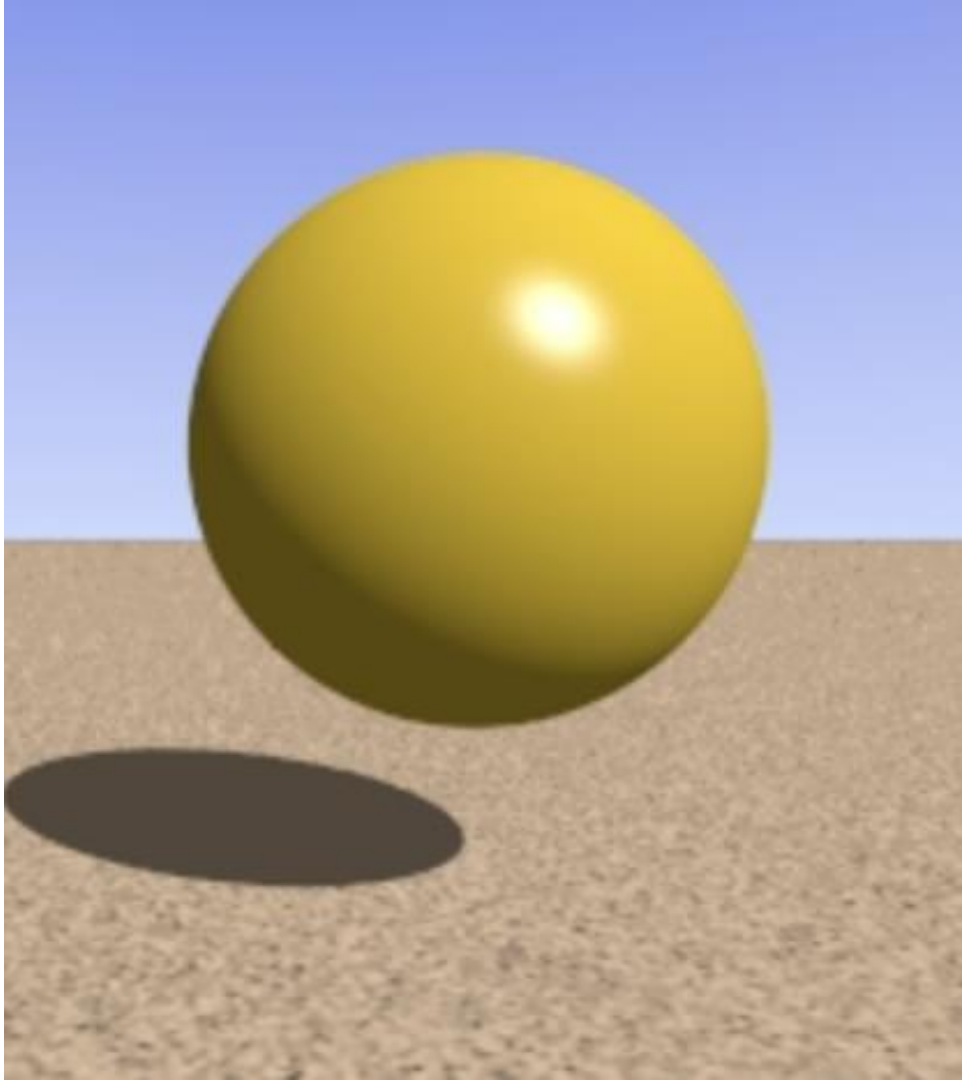
```
// Sets a box
// pigment: Sets a color for the box ("Red" as defined in
"colors.inc")
// finish: Sets how the surface of the box reflects light
// normal: Sets a bumpiness for the box using the "agate" in-
built model
// rotate: Rotates the box
 box          { <-0.5, -0.5, -0.5>,
                <0.5, 0.5, 0.5>
                texture { pigment { color Red }
                          finish  { specular 0.6 }
                          normal  { agate 0.25 scale 1/2 }
                        }
                rotate <45,46,47> }
```



Starting with POVRAY

- The Persistence of Vision Raytracer

Wikipedia example image from Gilles Tran



Blender

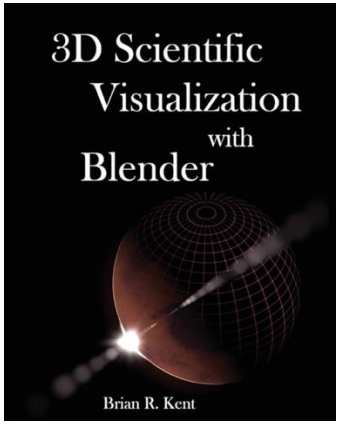
- Free and open-source software (Windows, mac, Linux …)
- Used for
  - Animated films
  - Visual effects
  - 3D-printed models
  - 3d applications
  - Virtual reality
  - Video editing
- Produced Academy-award-winning film Flow (2024)

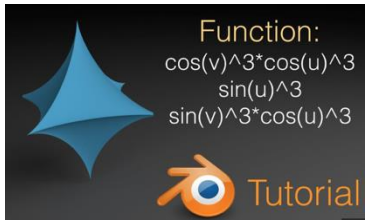
Available on Qantas inflight entertainment at the moment

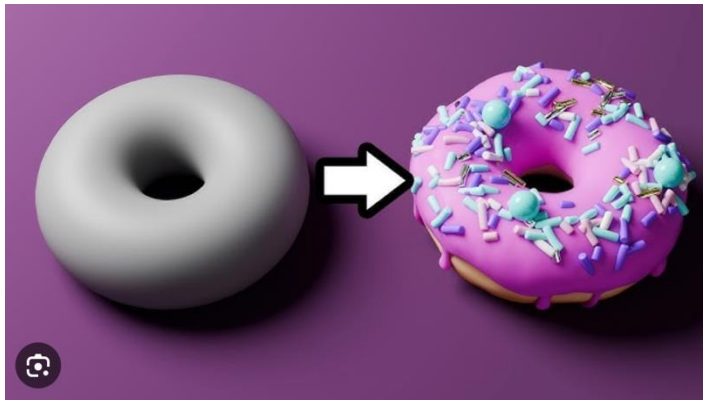
International theatrical release poster

Blender

- Has huge number of tools for
  - 3D modelling
  - 2D animation
  - Sculpting
  - Video editing
- Can model
  - Smoke
  - Rain
  - Dust
  - Cloth
  - Fluids
  - Hair
  - Rigid bodies
- Huge community – lots of YouTube tutorials





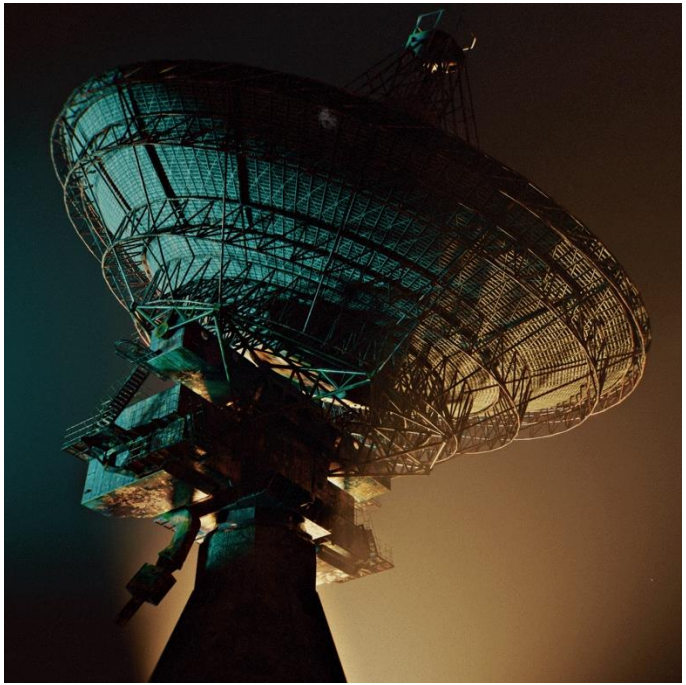


Images from Google image search

What I can’t do

Image from the film Flow

What I also can’t do ….



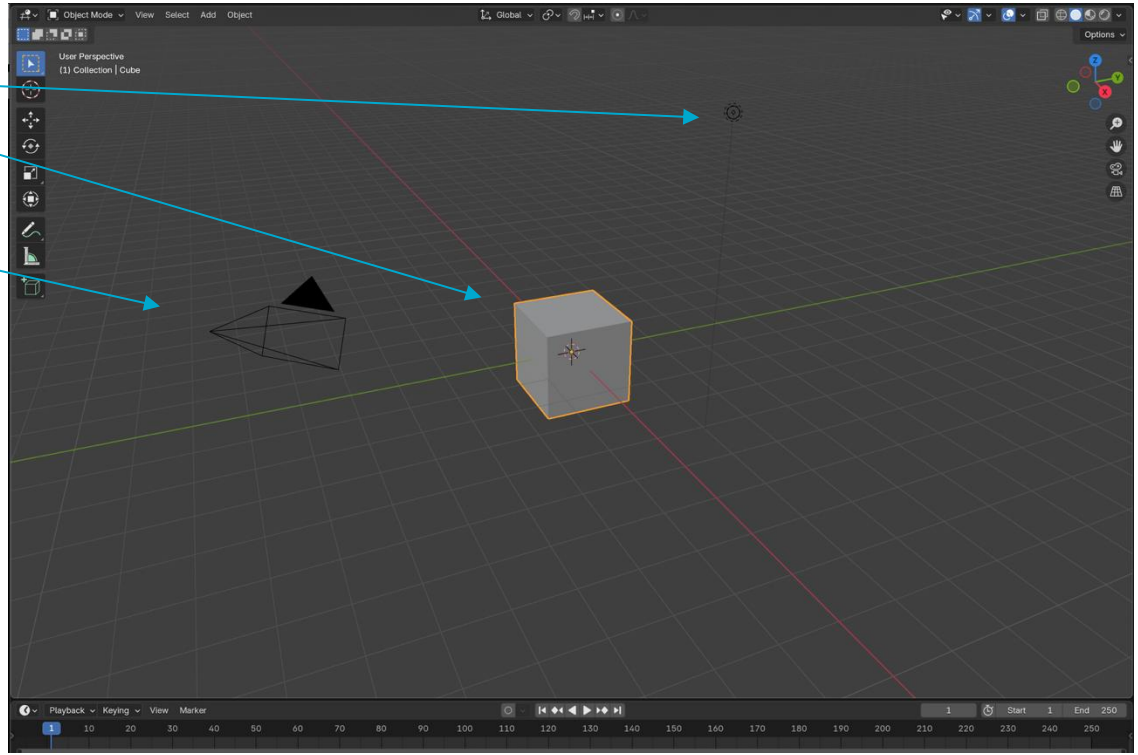
https://blenderartists.org/t/the-radio-telescope/1233674

What I can do … (takes about 30 minutes and following a YouTube tutorial)

Runs on my mac laptop with no extra tools (i.e. trackpad, normal laptop use …)

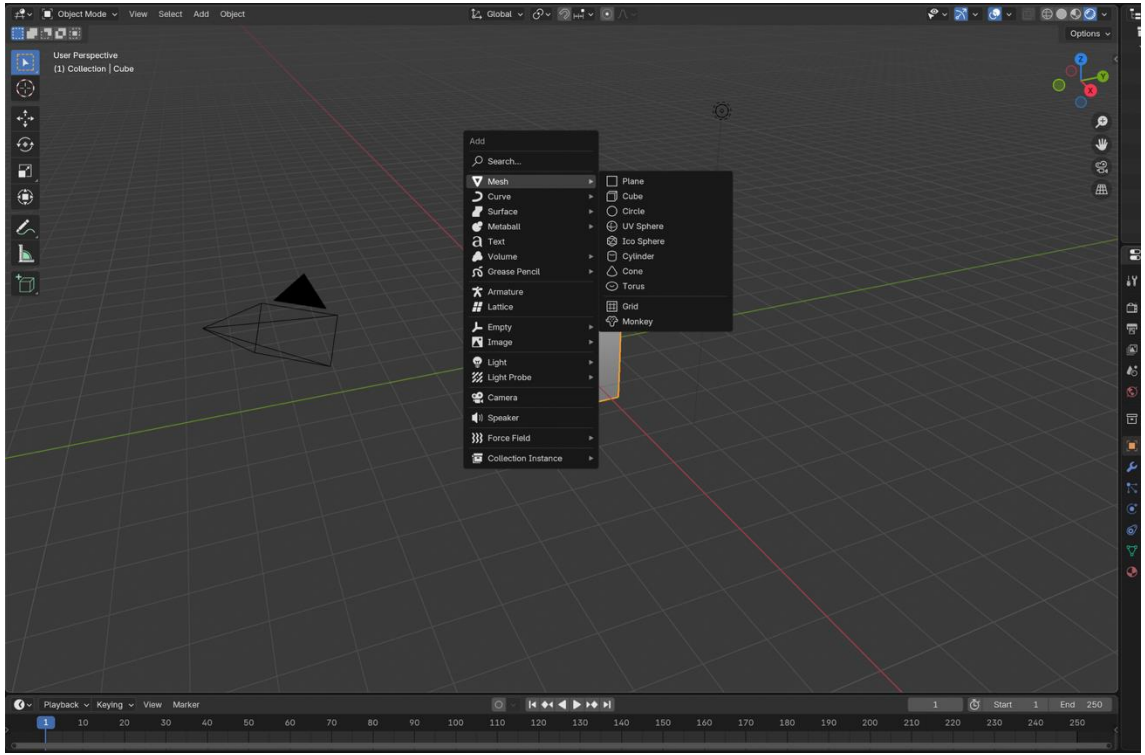
Basic ideas

- A light source
- An object in a 3D world
- A camera

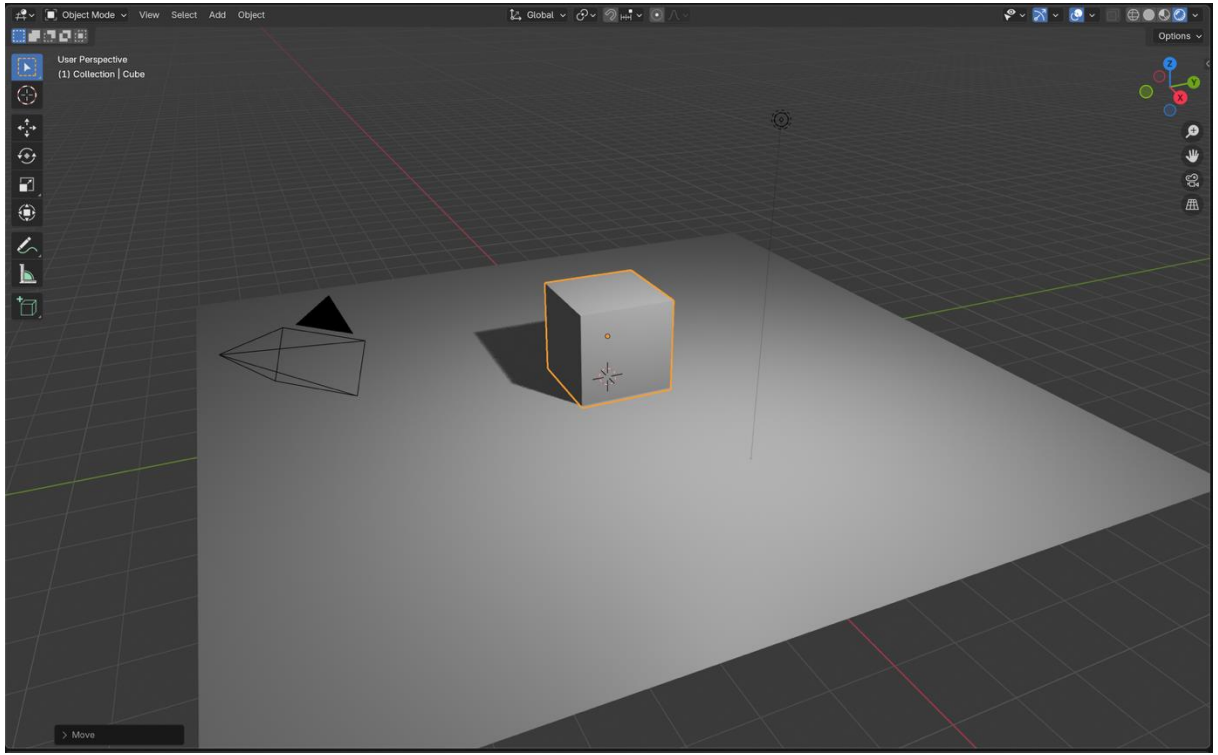


Time (for animation)

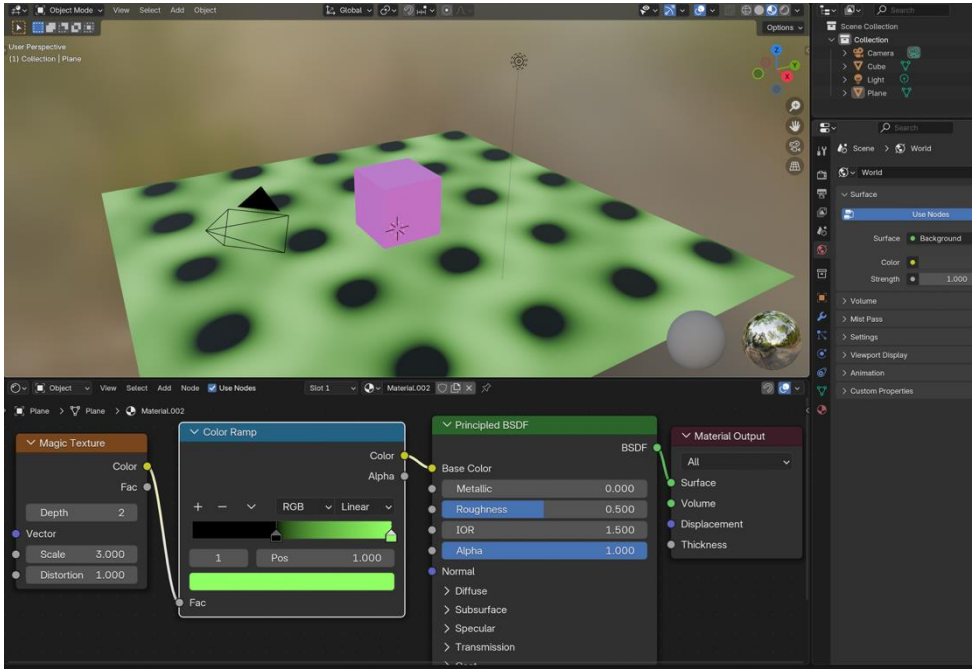
Basic ideas

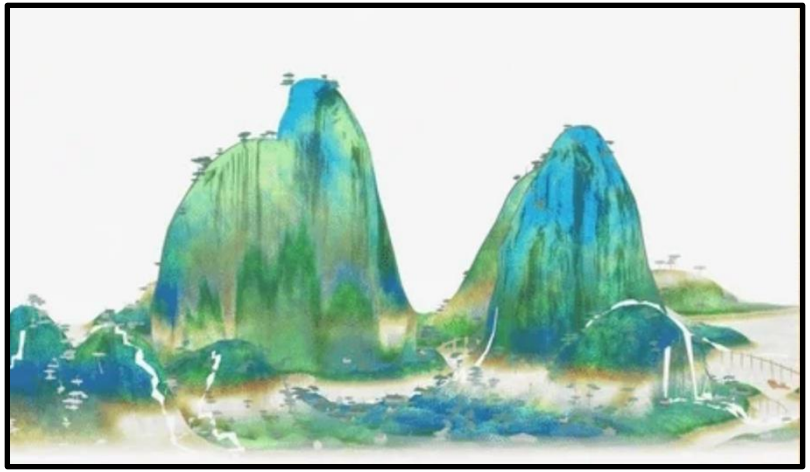

Object Mode
View
Select
Add
Object
Global
Options
User Perspective
(1) Collection | Cube
Add
Search...
Mesh
Curve
Surface
Metaball
Text
Volume
Grease Pencil
Armature
Lattice
Empty
Image
Light
Light Probe
Camera
Speaker
Force Field
Collection Instance
Plane
Cube
Circle
UV Sphere
Ico Sphere
Cylinder
Cone
Torus
Grid
Monkey
Playback
Keying
View
Marker
Start
End

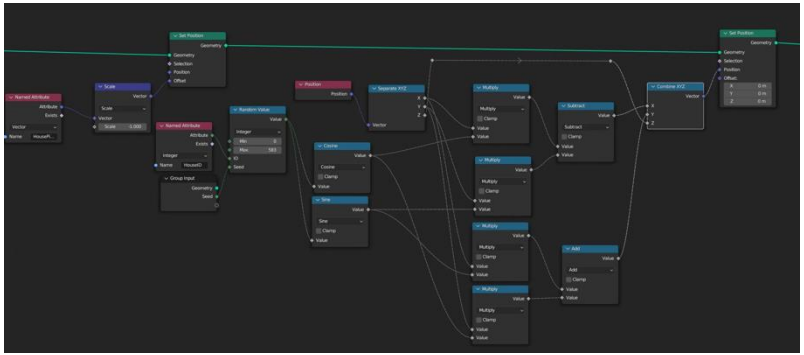
Basic ideas


Object Mode
View
Select
Add
Object
Global
Options
User Perspective
(1) Collection | Cube
Move

Basic ideas







https://shahriyarshahrabi.medium.com/procedural-chinese-landscape-painting-in-geometry-nodes-blender-3d-6417403430e6

Quick demo

Modelling some CASATTA-style antennas in Marsfield and on an SKA patch

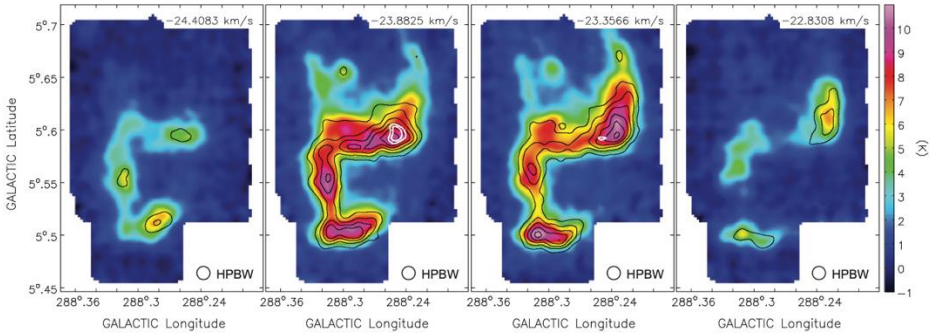
Original 3D design of CASATTA antenna from Maral Ansari

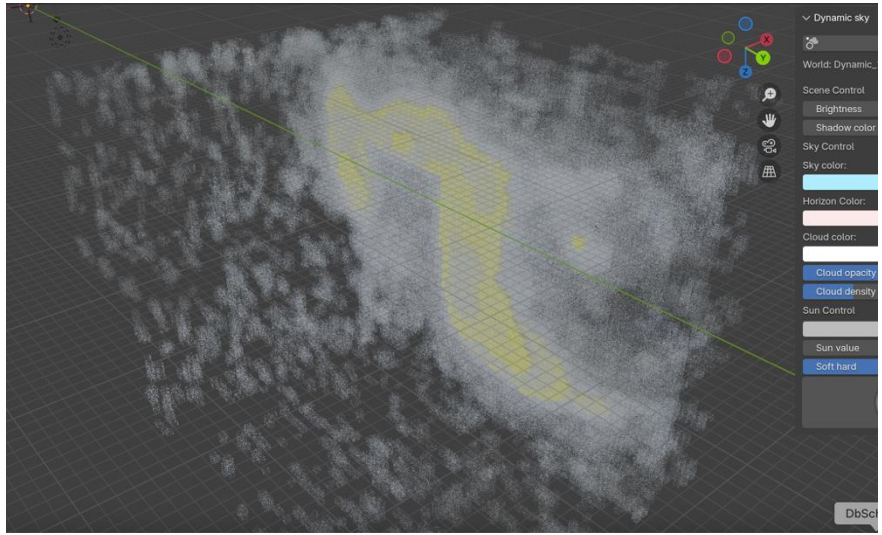
A 3D spectral line cube

Python inside of Blender:

```
coll = bpy.data.collections['Collection']


# Read data file
cube =
SpectralCube.read("cld2_combined_12co_imcombmine_bin2_bin3_2_2pixtrim.fits")
...
new_mesh = bpy.data.meshes.new('data')
new_mesh.from_pydata(verts,edges,faces)
new_mesh.update()
new_object = bpy.data.objects.new('graph',new_mesh)
mat = bpy.data.materials.new(name="Material")
new_object.data.materials.append(mat)
mat.diffuse_color = (1,0,0,0.0001)
coll.objects.link(new_object)
...
```





Data set from Jo Dawson. Cloud 2, in Dawson et al. (2011)

Not just me

ELSEVIER

Astronomy and Computing

Volume 51, April 2025, 100927

Full length article

FRELLED Reloaded: Multiple techniques for astronomical data visualisation in Blender

R. Taylor [1] ✉

Show more

+ Add to Mendeley Share Cite

https://doi.org/10.1016/j.ascom.2024.100927 ↗ Get rights and content ↗

Abstract

I present version 5.0 of FRELLED, the FITS Realtime Explorer of Low Latency in Every Dimension. This is a 3D data visualisation package for the popular Blender art software, designed to allow inspection of astronomical volumetric data sets (primarily, but not exclusively, radio wavelength data cubes) in real time using a variety of visualisation techniques. The suite of Python scripts that comprise FRELLED have been almost completely recoded and many new ones added, bringing FRELLED's operating environment from Blender version 2.49 to 2.79. Principle new features include: an enormously simplified installation procedure, a more modular graphical appearance that takes advantage of Blender 2.79's improved interface, much faster loading of FITS data, support for larger data sets, options to show the data as height maps in 2D mode or isosurfaces in 3D mode, utilisation of standard *astropy* and other Python modules to

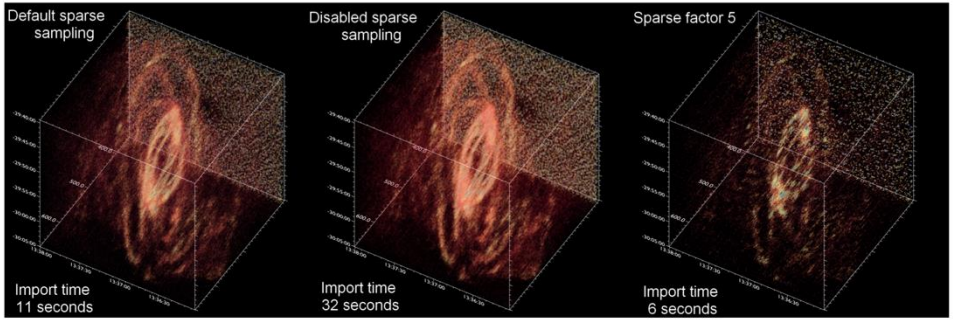


Figure 2: Illustration of the different image quality with and without different levels of sparse sampling, using the THINGS HI data cube for NGC 5236 (Walter et al. 2008). Using either the default (left) or disabled (middle) sparse sampling results in no appreciable difference in image quality, except when zooming in, for a difference in loading times of almost a factor of three. Using a high sparse factor (right) gives faster previews (considerably faster, depending on the data set) but at the expense of quality. The import times quoted are for all three projections of the data, hence the non-linear values of import speed as the sparse sampling does not affect all projections equally; additionally, this neglects the 13 seconds required in each case to load the FITS file into memory and generate the axes.

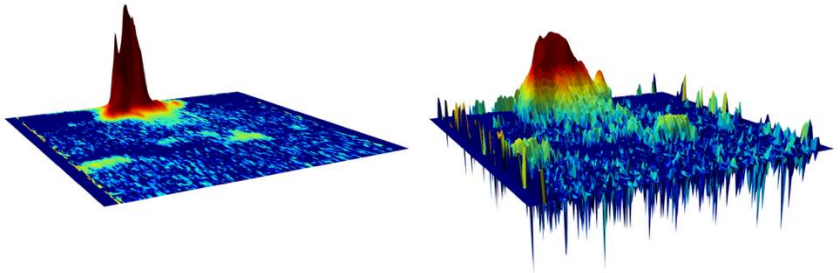
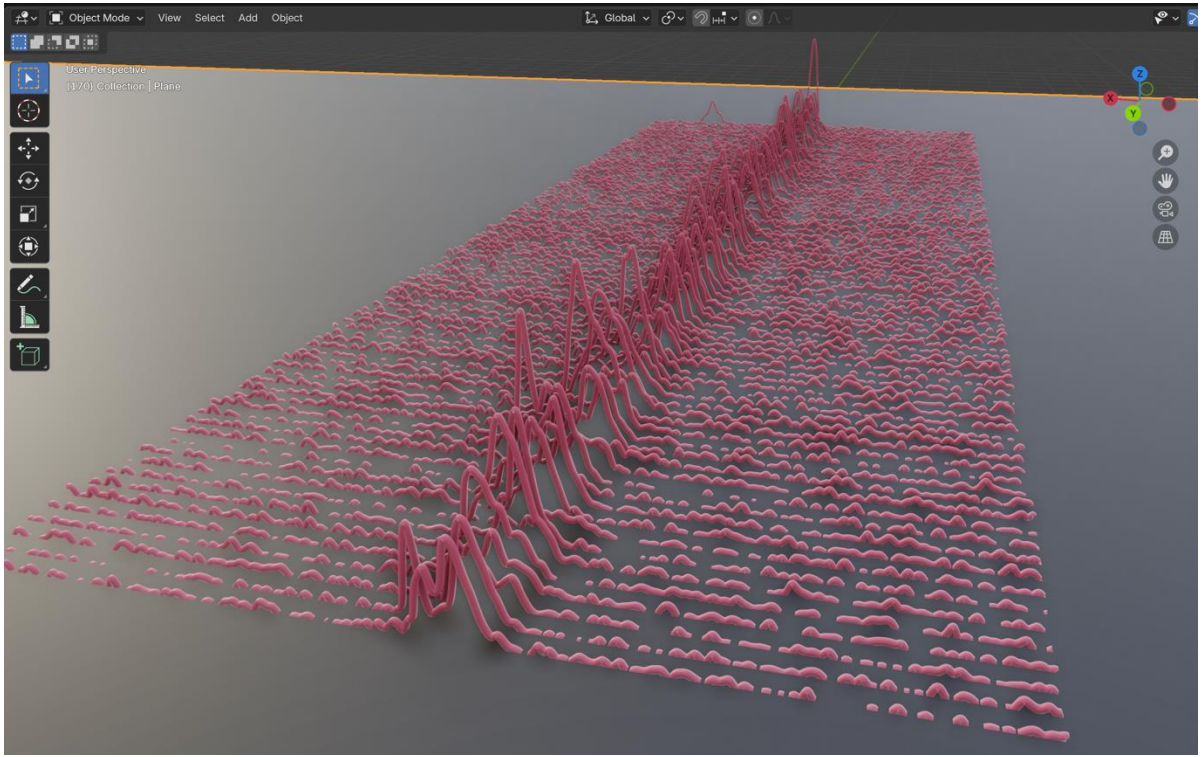


Figure 4: Height maps of a position-velocity slice of the supplied M33 data cube using arbitrary scaling. The left panel uses the default linear scaling while the right panel using logarithmic scaling for the displacement.

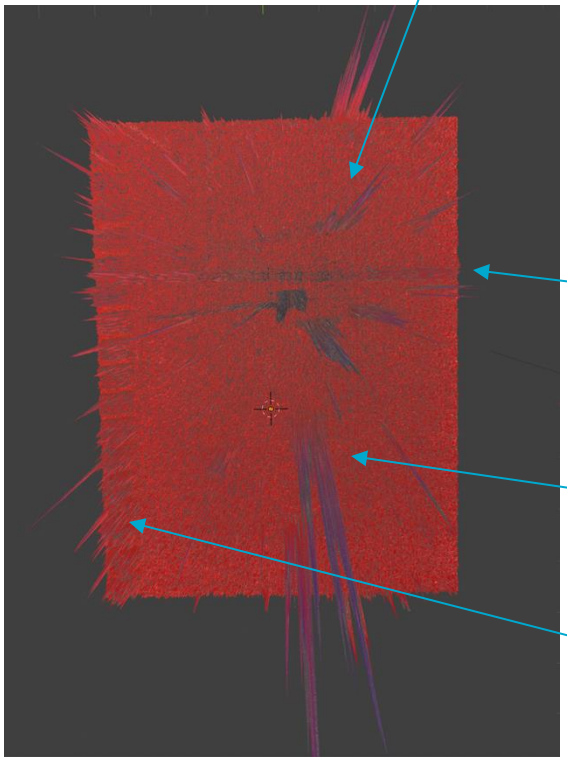
Pulse trains

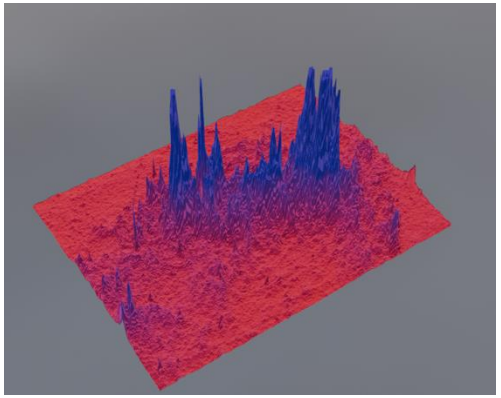
- Using python to load in a pulse train from a pulsar observation using Murriyang.



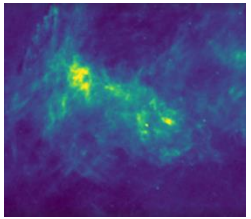
From Murriyang's P1161 data set

Colour maps

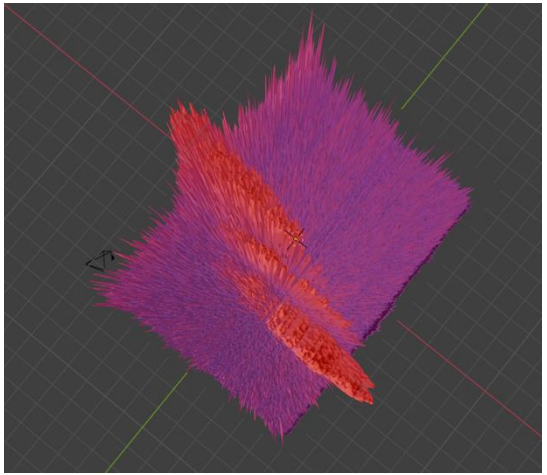


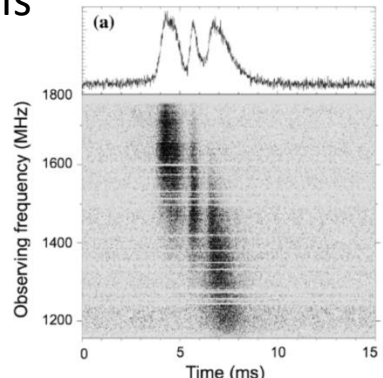


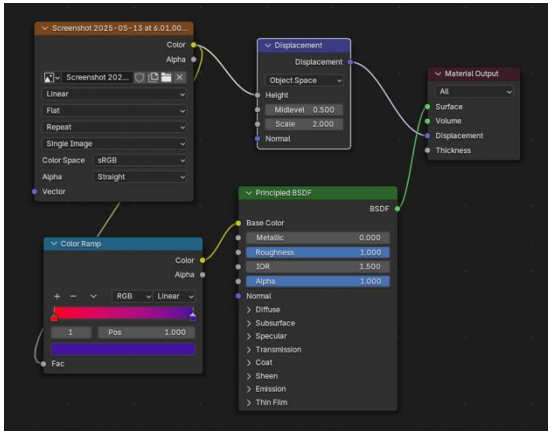
Mradumay's gas cloud



FRB 121102





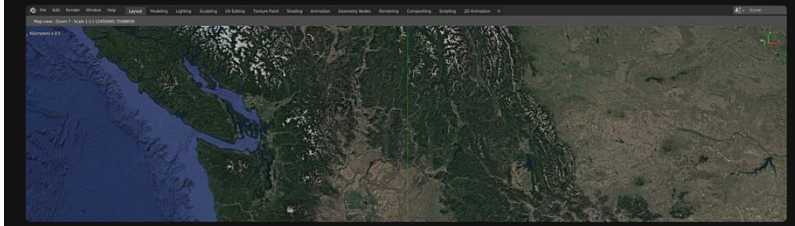


Modifying videos

With thanks to Wenhua Ling

Extending Blender

BlenderGIS (Free)



I've opted not to talk about super niche import/export formats here, but BlenderGIS (GIS standing for Geographic Information System) is unique because it allows you to browse and use satellite maps directly in Blender. To use it, just browse for the area that you're looking

X-Muscle System ($35)

Graswald (Free Limited or $10/Month Subscription)

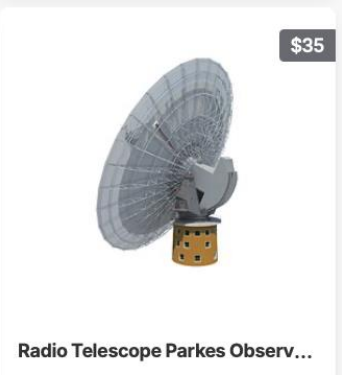
Khaos ($35)

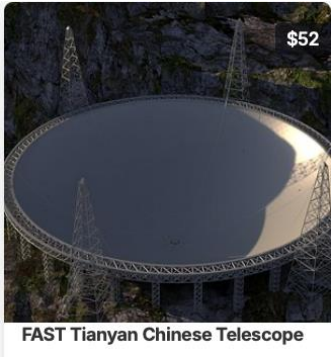
The Khaos add-on also helps with rigid body destruction (and comes with several nice packs of photo scanned debris), but it also has a big focus on creating explosions via Blender's particle and smoke systems.

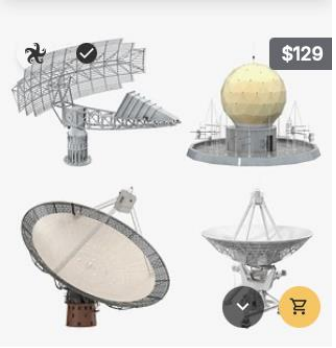
MattePainter ($29)

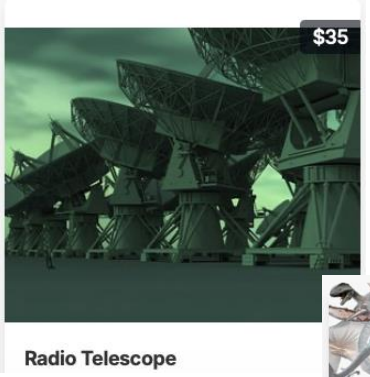
https://cgcookie.com/posts/the-ultimate-guide-to-the-best-blender-add-ons

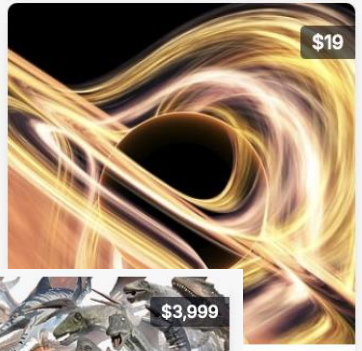
CSIRO











- Murriyang is very popular in the 3D-model shops online! (https://www.turbosquid.com)

Can we model radio waves?

- Probably not (not modelling EM-waves), but …

A Blender-based channel simulator for FMCW Radar

Yuan Liu, Moein AHMADI, Johann Fuchs, Mohammad Alaee-Kerahroodi, M. R. Bhavani Shankar,
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855, Luxembourg
Email: {yuan.liu, moein.ahmadi, johann.fuchs, mohammad.alaee, bhavani.shankar }@uni.lu

***Abstract*—Radar simulation is a promising way to provide data-cube with effectiveness and accuracy for AI-based approaches to radar applications. This paper develops a channel simulator to generate frequency-modulated continuous-wave (FMCW) waveform multiple inputs multiple outputs (MIMO) radar signals. In the proposed simulation framework, an open-source animation tool called Blender is utilized to model the scenarios and render animations. The ray tracing (RT) engine embedded can trace the radar propagation paths, i.e., the distance and signal strength of each path. The beat signal models of time division multiplexing (TDM)-MIMO are adapted to RT outputs. Finally, the environment-based models are simulated to show the validation.**

***Index Terms*—Blender, channel simulation, FMCW radar, indoor pedestrian, ray tracing.**

I. Introduction

Radio-based sensing systems have long and widely been used for various radar applications. A basis for designing and optimizing sensor systems is the knowledge of radar channel characteristics [1]. Conventionally channel models can be obtained by field measurements. However, measurement can be time-consuming and also expensive[2]. The deterministic ray tracing (RT) [3, 4] has been used in wireless communication simulations. Following the trends, recent studies utilize the RT tool embedded in the animation software, e.g., Blender and Optix, for radar channel simulation [5]. The Blender and

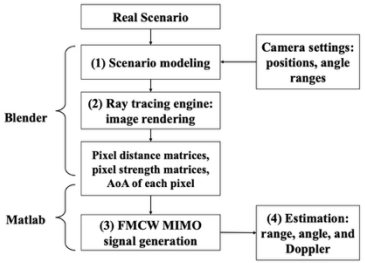


Figure 1. Outline of the simulation procedure.

Considering a M-Tx and N-Rx radar array. The beat signal of

Mon. Not. R. astr. Soc. (1992) **258,** 217–224

Complex, off-axis sidelobes of a radio telescope caused by feed-support legs

Andrew Hunt and Alan Wright
Australia Telescope National Facility, CSIRO, PO Box 276, Parkes, NSW 2870, Australia*

Accepted 1992 March 6. Received 1992 February 26; in original form 1991 December 20

3 AN EXPERIMENT

We decided to test this theory with a practical experiment. Our visitors' centre at the Parkes Observatory has a large-scale model of the dish. We fitted a semi-cylindrical mirror to the underside of the model's lower leg, and shone a strong electric torch on to this to simulate the Sun.

At once an important effect became apparent: the torch beam was scattered from the leg into a cone which reached the dish surface *as an arc* rather than as a spot of light. Some of this light was incident on the dish in a direction parallel to the principal axis and would thus be reflected into the feed. Conversely, we realized that if the light source lay anywhere on the surface of a circular cone about the leg, some portion of it would reach the feed.

This was confirmed by moving the torch, and we found

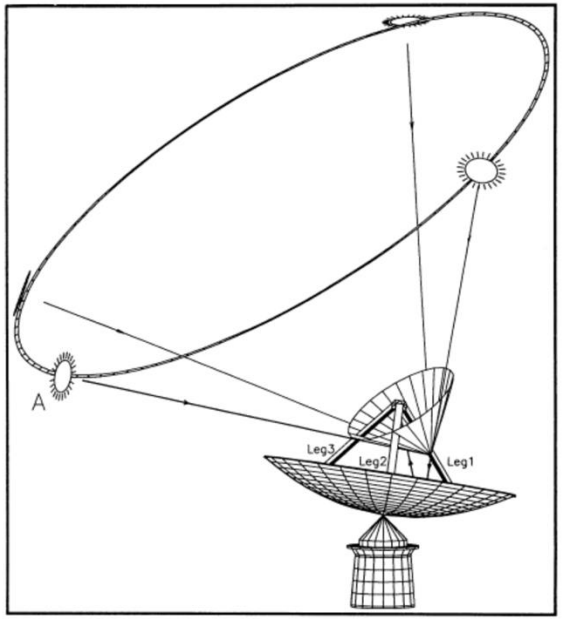
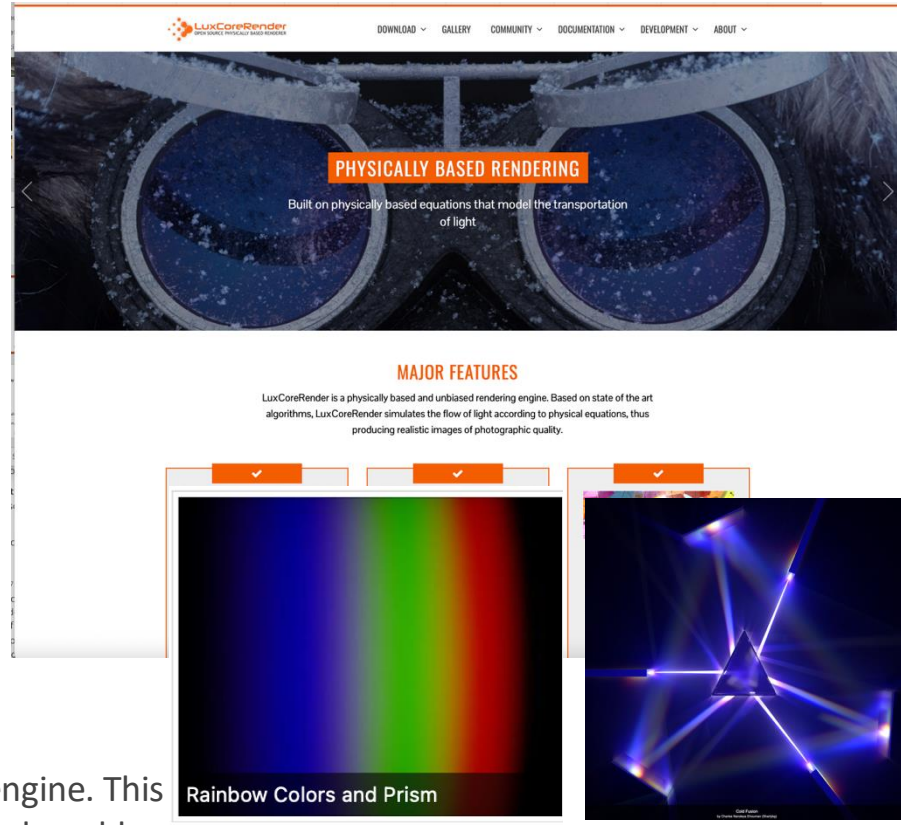
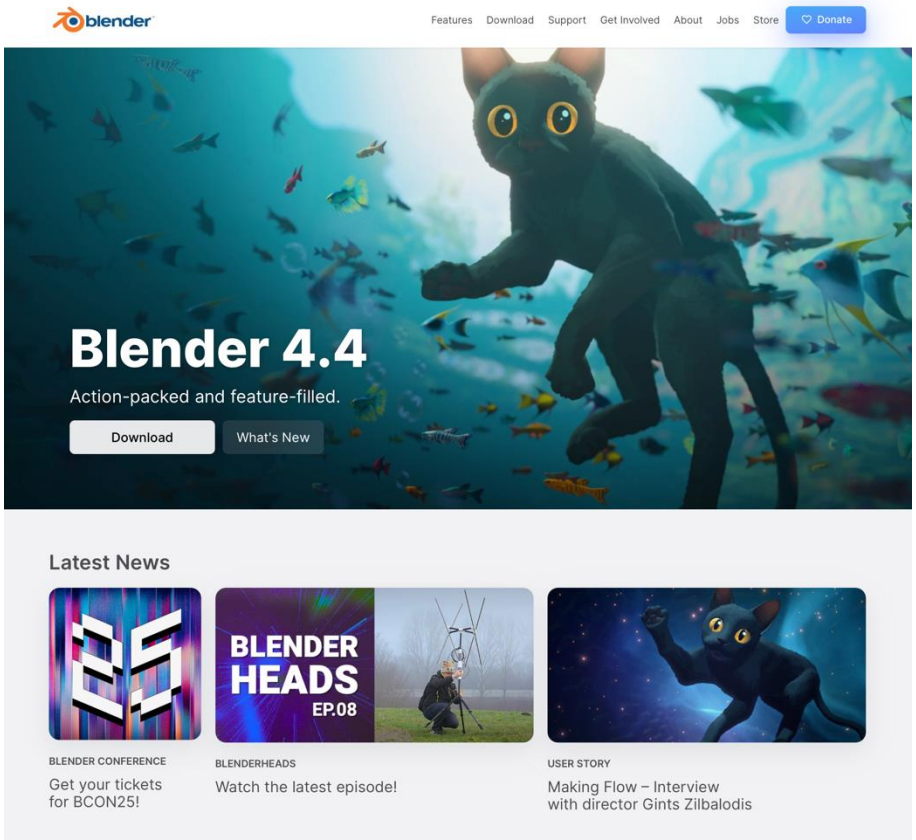


Figure 3. This figure shows the annular, conical sidelobe about the lower feed-support leg (Leg 1), together with the path of a single ray from the Sun (at position 'A') which is reflected from its underside. This ray then travels parallel to the principal axis of the dish and is, finally, reflected by the dish surface into the feed. To enter the feed, the complement of the angle of incidence of the ray on the leg (i.e. the half-angle of the cone) must equal the angle of inclination (γ) of the leg to the principal axis.



LuxCoreRender is a physically correct, unbiased rendering engine. This means that LuxCoreRender does not use tricks to imitate real world behavior: all calculations are done according to mathematical models based on physical phenomena. In LuxCoreRender we will always make the 'unbiased' design choices.

https://www.blender.org



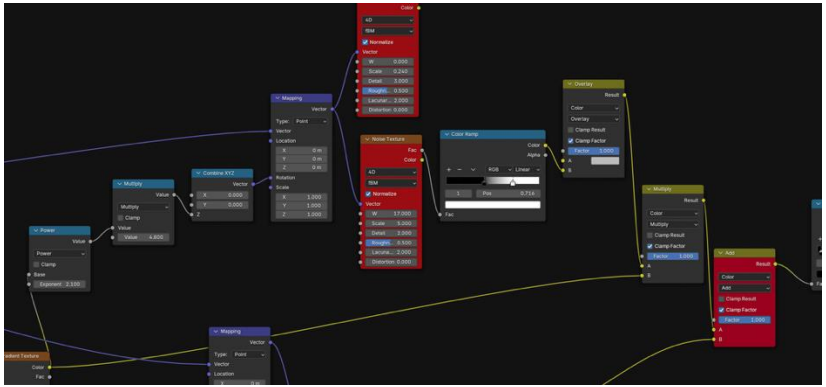
- Highly recommend a play
  - Easy to download and get started
  - Free
- Amazingly powerful
- Hugely fun
- **… but …**
  - Can it be scientifically useful? Not sure

https://www.ursi.org/proceedings/procGA21/papers/URSIGASS2021-Sa-J11-AM2-3.pdf

https://www.sciencedirect.com/science/article/abs/pii/S2213133724001422

Thank You

george.hobbs@csiro.au



Based on: https://www.youtube.com/watch?v=HCBW4fNAjBg