Perceiving Real Depth on Flat Screens with Cross Eyed 3D

Yashodhan M.

28 July 2025

Abstract

In the field of visualisation, we are limited by the flatness of our screens and paper. Advancements are being made to evade this limitation.
In this project we explore the possibility of encoding depth information with parallax in computer visualisations. We discuss the method, the development pipeline, a lab study, other work and criticisms.

1 Principles of Visualisation

- Visualisation is to augment human abilities; to aid humans understand, navigate, find patterns in data.
- New algorithms and methods of visualisation are justified when they help us carry out our tasks more effectively.

1.1 Important Considerations for 3D Visualisation

- We do not really perceive the world in 3D. Not more than a creature living in "Flatland" can see its world in full 2D.
- 3D visualisations should be avoided when they confuse the user more than they help; when 2D visualisations are sufficient and when some of the data gets obscured.
- Because of the 3D perspective, marks in the deeper part of the visualisation become smaller and hence harder to discern.

2 The Problem and Proposed Solution

2.1 The Problem

Two dimensions is simply insufficient in conveying the information contained in the scientific publications today. We often create multiple plots, encode a third dimension with colour or other channels and employ other methods to escape the limitations of our flat screens.

In addition, since we have depth perception in real life, a solution allowing us to perceive depth in images would match our worldly experience more closely.

2.2 The Proposed Solution

In this project we plan to use stereoscopy, in particular cross eyed 3D method, to add a third dimension to our flat visualisations.

The **depth** would be encoded with **parallax**.

3 Introduction to Stereoscopy

3.1 What is Stereoscopy?

Definition 1 (Stereoscopy) *Stereoscopy is the general scientific and technical approach to creating or enhancing the illusion of three-dimensional depth from two-dimensional images. It relies on presenting slightly different views of the same scene to each eye, mimicking natural binocular vision. This technique is used in various applications such as 3D photography, virtual reality, and computer graphics.*

3.2 What is Stereography?

Definition 2 (Stereography) *Stereography refers to the process or method of creating stereoscopic images or drawings. It involves generating a pair (or more) of images from slightly different perspectives that can be viewed together to produce a perception of depth. The term encompasses both the design and production techniques used to create these visualizations.*

Definition 3 (Stereogram) *A stereogram is the resulting image or set of images produced through stereography. It typically consists of two or more related images intended to be viewed simultaneously with one image for each eye. When viewed correctly—using techniques like crossing the eyes or using special glasses—the brain combines the images into a single three-dimensional perception.*

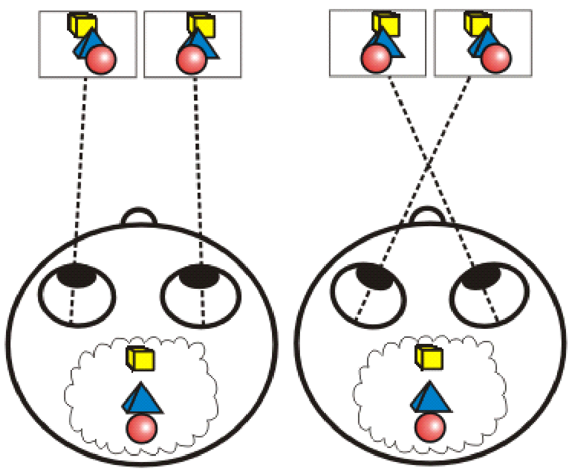

Figure 1: How the cross eyed 3D method works (TRIAXES, 2025).

3.3 How to view cross eyed 3D

It would help a viewer to understand how we cross our eyes to focus on different objects.
To view a cross eyed 3D stereogram, the viewer must cross their eyes until they see the two images become four, and then combine the images in the middle so they have three images. Then wait for your eyes to lock in focus on the middle image. It may take a little bit of trying to get it right on the first attempt.
You may also use the "pinky method", where you touch your nose with your thumb and extend your pinky out and focus on it. Lower the pinky while keeping your eyes in that position.

3.4 Historical Context

Actually, stereoscopy is almost as old as normal photography, which one might call monoscopy. Find below2 a photograph from 1860 where the user is inteded to use cross eyed 3D to view the street in 3D.
The technique is as easy as placing the camera in two places and capturing the same scene. One could also place two cameras separated by a distance. You do not have to place the cameras exactly as far apart as your eyes. You are even free to put the cameras a kilometer away, doing this you would allow the viewers to perceive depth in faraway mountains which they otherwise not be able to with just their unaided eyes.

Figure 2: A photograph taken in 1860 (Soule, 2025).

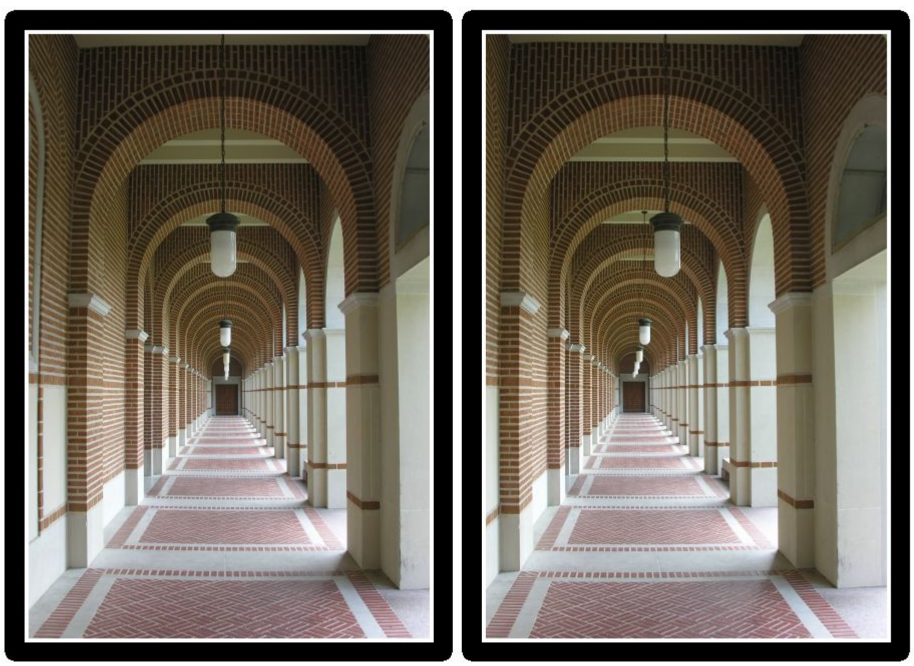

Figure 3: A stereophoto of a passage. Credit: digital-photography-school.com

Stereoscopy in academia is also not new!
Here you can find an example from Methods of Theoretical Physics by Morse and Feshbach, which was hand-drawn!

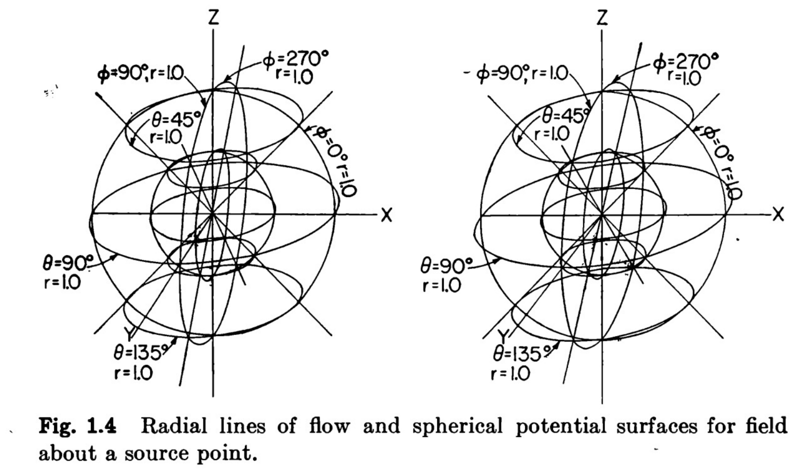

Figure 4: A mathematical diagram from (Morse and Feshbach, 1953).

3.5 Practical Demonstration

For this project, we also took images of the hall outside the PhD student offices and found out that stereoscopic photography was quite simple to do and there is not a lot of chance of failure if you are moderately careful.
You can also observe how increasing distance between camera positions affects depth.

(a) 1 x eye separation

(b) 2 x eye separation

Figure 5: Stereo image pairs demonstrating different camera separations.

4 Development Through The Nested Analysis Framework

Now let us come to the topic of developing a program to create cross eyed 3D visualisations taking into consideration the following nested analysis framework.

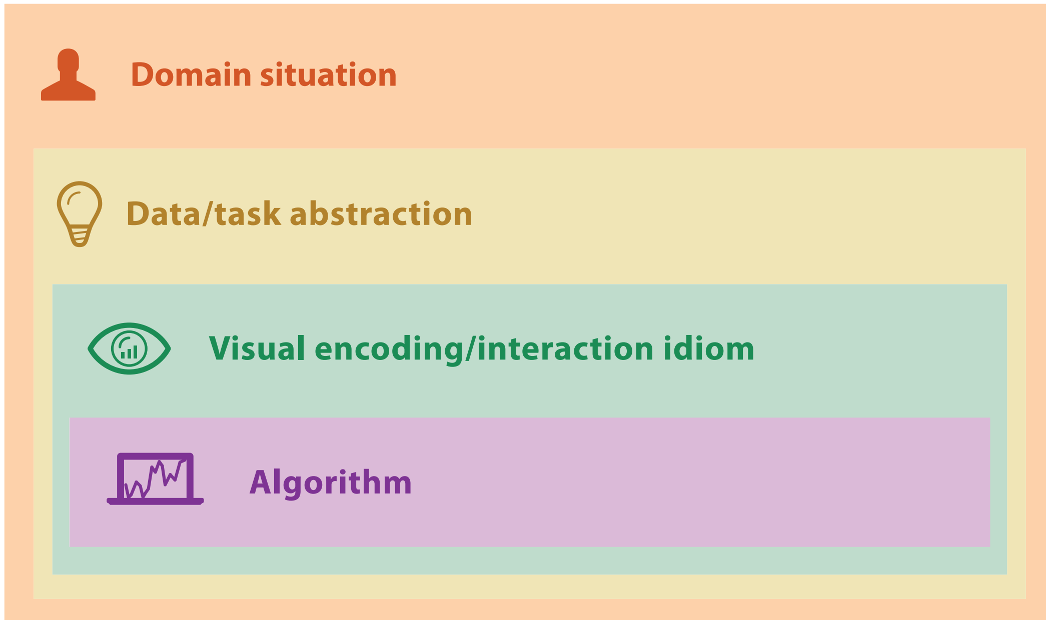

Figure 6: The nested analysis framework by (Munzner, 2009).

4.1 Domain Situation

Since the scope of this visualisation method is vast, the target users range from school children to professional scientists.

As we shall find out later, the subset of people who would be able to use this visualisation is not determined by profession, age, gender, nationality etc.

The visualisations generated in the project are more oriented towards research and education; hence the target audience for these particuliar stereograms are students and academics.

4.2 Data and Task Abstraction

- **What** is shown? A range of discrete and continuous dataset types such as items, positions, networks.
- **Why** is the user looking at it? For a range of possible actions such as to enjoy, to discover, to search or to analyse.
- **How** is it shown? Through a stereogram, particularly cross eyed 3D.

4.3 Visual Encoding

4.3.1 Why Stereoscopy?

- An extra dimension could be visualised.
- The visualisation could be easily printed on paper and scaled to any size.
- Besides scientific plotting, images with depth are more aesthetically pleasing.

4.3.2 Choice of Visual Encoding

Parallel eye viewing was considered as an alternative to the cross eyed method, but cross eyed method was chosen as parallel eyed method only works when the distance of the eyes to the screen is just right for a particular size of the image whileas the cross eyed method allows flexibility over viewing distance.

Anaglyphs (Red and Blue 3D glasses) and polarised glasses were not considered as they are an additional requirement.

4.3.3 The Visual Encoding

The visual encoding is simply two images or plots next to each other, the left one corresponding for the right eye's vision and the right one corresponding to the left eye's vision.

4.4 Algorithm

4.4.1 The Choice of Software

- We used the Python programming language for this task.
- We used NumPy (Harris et al., 2020) library in python for many mathematical functions and data structures.
- The program is built on the Matplotlib (Hunter, 2007) library in python.

4.4.2 Program Algorithm

Coordinates x, y, z, and optional parameters for styling and layout
Output: A cross-eyed stereo visualization (two subplots)

1. **Input Processing -** Convert input arrays x, y, z to NumPy arrays. - Handle surface or scatter inputs by flattening as needed. - Validate dimensions and ensure all $z > 0$.
2. **Parallax Displacement Calculation -** Compute horizontal displacement for left and right views:

   $$x_{\text{left}} = x - \frac{\delta}{z}, \quad x_{\text{right}} = x + \frac{\delta}{z}$$

   where δ is the parallax factor.
3. **Figure Setup -** Create a figure with two subplots if not provided. - Set background color, remove ticks, and adjust spacing.
4. **Axis Configuration -** Set axis colors and borders based on input. - Optionally disable spines (borders) around axes.
5. **Depth Sorting -** Sort points from farthest to nearest (based on z-values). - Reorder x, y, z coordinates accordingly.
6. **Dynamic Sizing -** Scale point sizes inversely with depth (s/z). - Adjust for sorting if applied.
7. **Color Mapping -** Use scalar or array of colors. - Apply sorting if needed.
8. **Scatter Plotting -** Plot points on both left and right axes using adjusted x-coordinates.
9. **Auto-scaling -** Automatically set axis limits with padding for visibility.
10. **Focus Dots (Optional) -** Add red dots at the bottom of the figure to help with alignment when viewing cross-eyed.
11. **Title (Optional) -** Display title on both subplots.

4.4.3 Computational Efficiency

For all the different types of plots created in this project (shown in the results section), the computation time centered around 75 milliseconds. The computer has 32GB of RAM, an Intel i7 processor and 12GB of GPU memory.
Since the time taken was small, no measures to improve perfomance were considered. Writing the code in Julia would have been the first step towards lowering computational cost.

4.5 Lab Study and Validation

A lab study was conducted with students and researchers connected to the author and the feedback was incorporated into the model used.

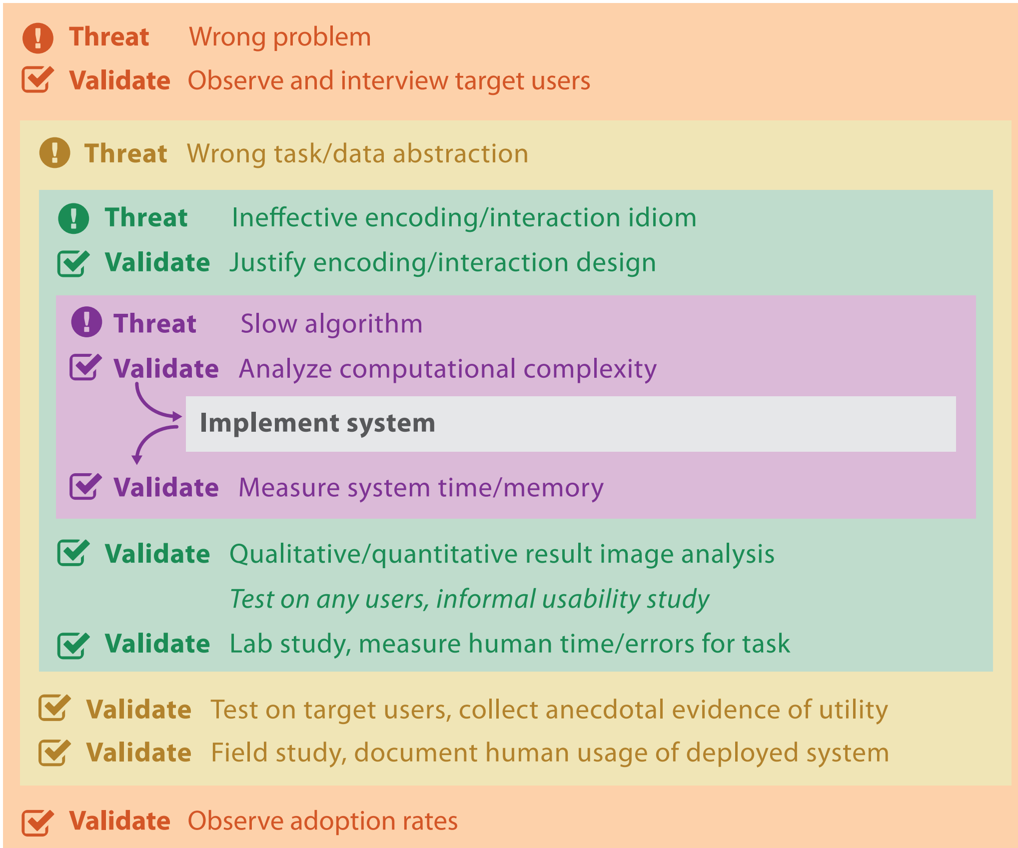

Figure 7: The validation process from (Munzner, 2014).

4.5.1 User Feedback and Improvements

- The first and the starkest observation was that not everyone could do this. Some could do it immediately, some after trying a little bit, some failed after a lot of trying and many would not try it. Additionally some people reported being unable to cross their eyes at all.
- Out of 11 users who were specially asked to try, 7 succeeded.
- Users who succeeded reported enjoying the visualisations.
- Users could correctly identify relative depths.
- Smaller sizes of the diagrams were reported to be easier to view.
- A user suggested plotting lines in 3D. He also suggested adding a red dot for guidance.
- A user suggested exploring sterescopic projections of 3D computer models.

5 Final Results

Below are the results of the program in many usecases. Note that other smaller programs were written for some of these visualisations in order to process or generate the positions needed to plot.

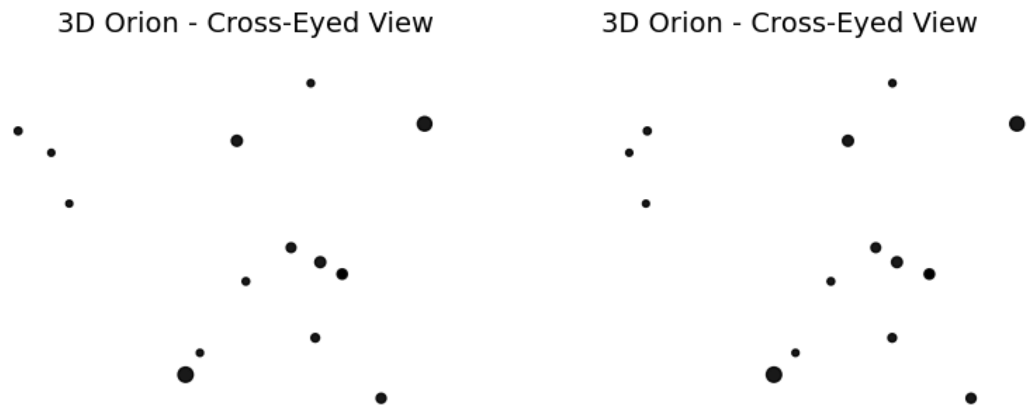

Figure 8: Orion constellation visualization.

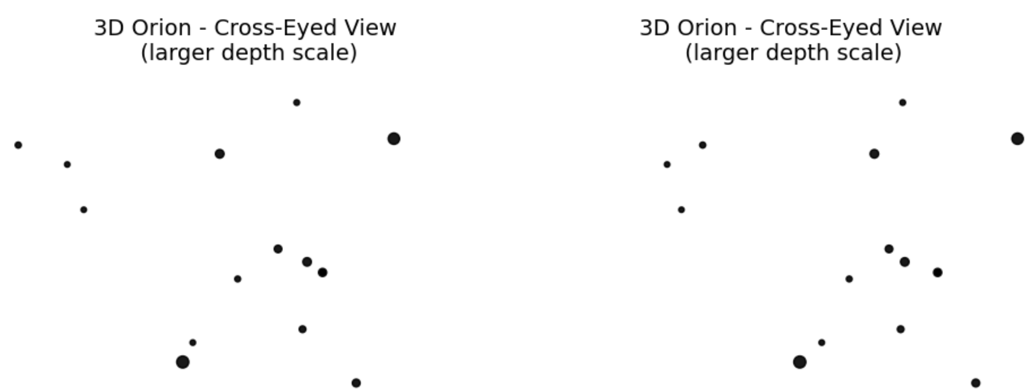

Figure 9: Orion constellation with exaggerated depth.

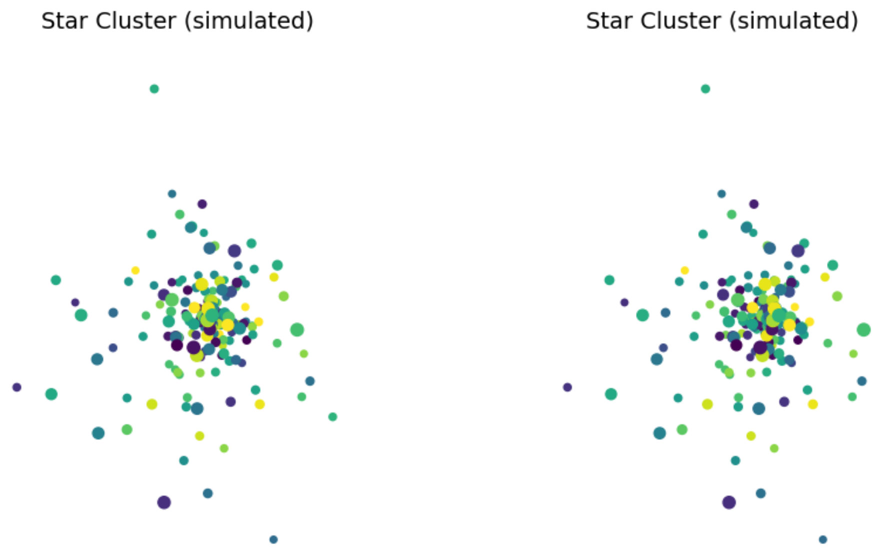

Figure 10: Star cluster visualization.

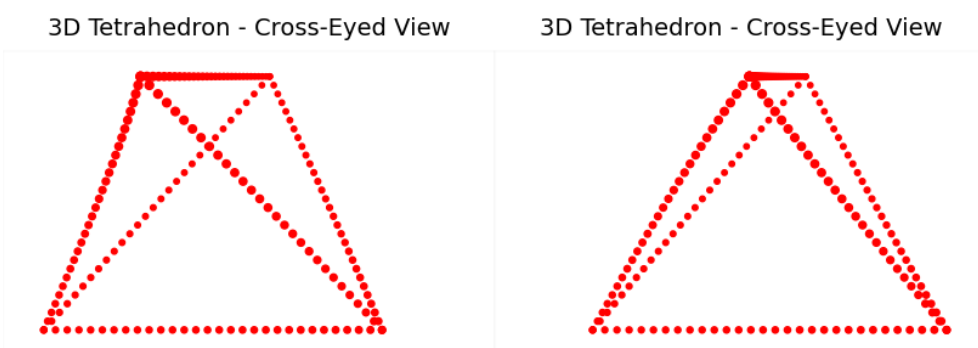

Figure 11: Tetrahedron visualization.

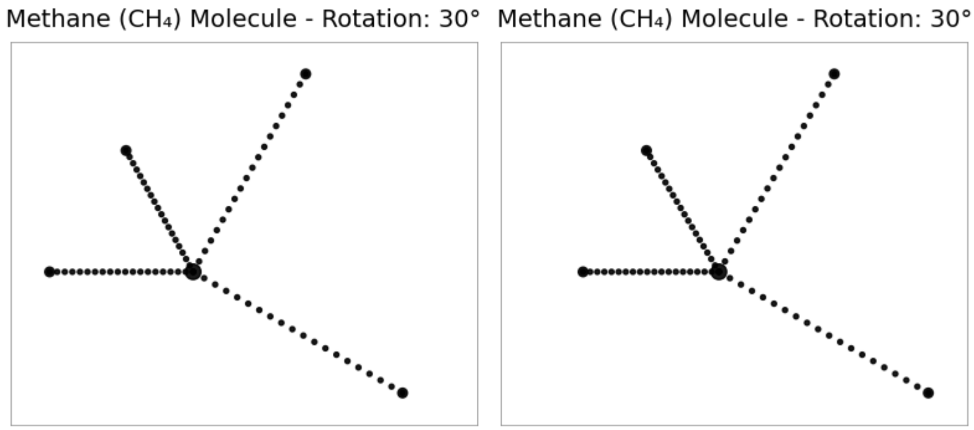

Figure 12: Methane molecule visualization.

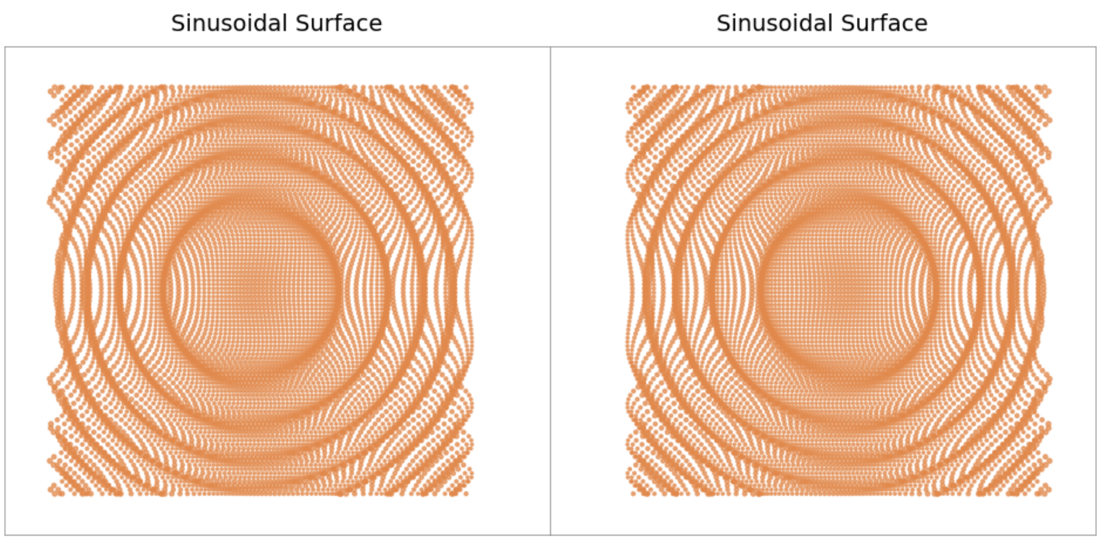

Figure 13: 2D sine wave visualization. Note that this is off, shown only to demonstrate that plotting surfaces is possible.

6 Applications, Future Work and Conclusion

6.1 Drawbacks

- **Focus:** We see depth by converging our eyes on the target, but we also have a lens in the eye which is a part of the depth perception system. Here to see clearly the user focuses on the flat screen while seeing objects appearing to be behind it. This prevents a complete depth experience.

- **Space:** Placing two plots in the same space means we sacrifice half of the allotted horizontal space. Such a diagram would be unpleasant in a two-column article.

- **X coordinate:** The X coordinates of the points in the plot are misleading. The real X coordinates are the averages of the two values seen.

- **The Z axis scale:** On a normal plot the Z axis scale inside the page, and hence becomes invisible. The axes must be rotated to see all three scales, and this functionality is not yet added.

6.2 The mpl stereo Library

A small library built on top of matplotlib was discovered towards the end of the project. It is created by Scott Shambaugh and serves a similar function.

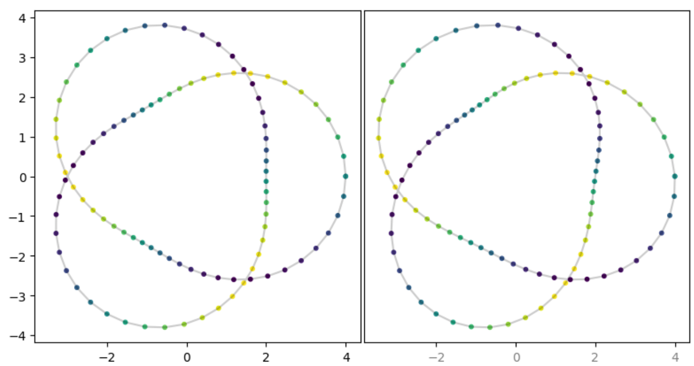

Figure 14: Trefoil knot stereogram using mpl-stereo (Shambaugh, 2024).

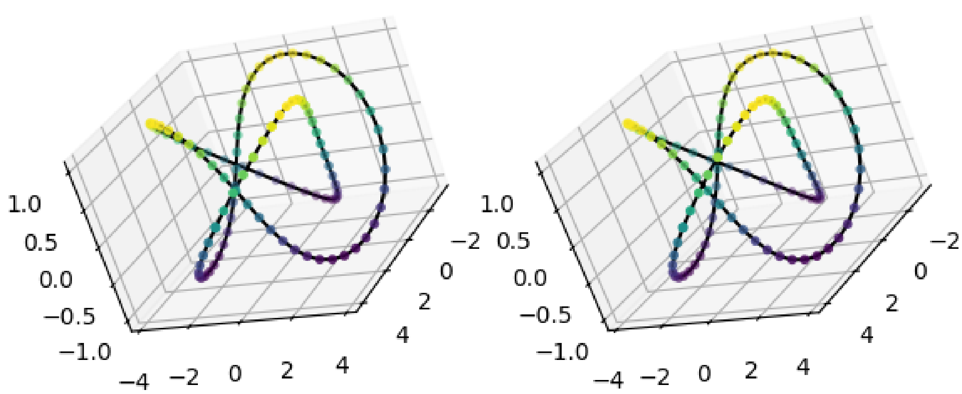

Figure 15: Trefoil knot 3D plot using mpl-stereo.

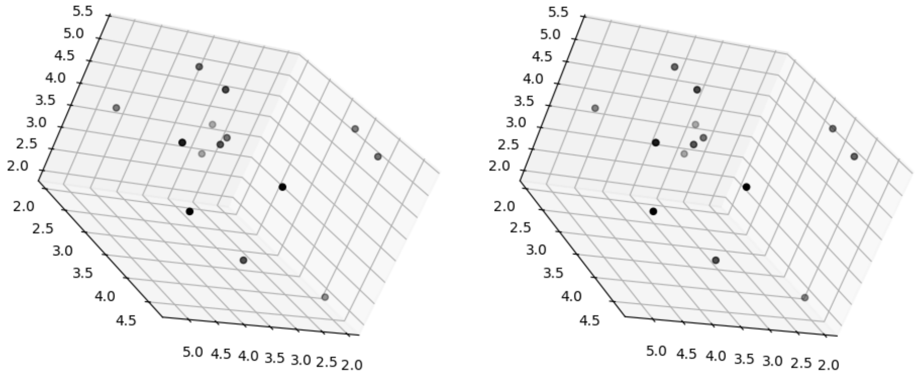

Figure 16: A 3D plot of the Orion constellation using mpl-stereo.

6.3 Community Examples

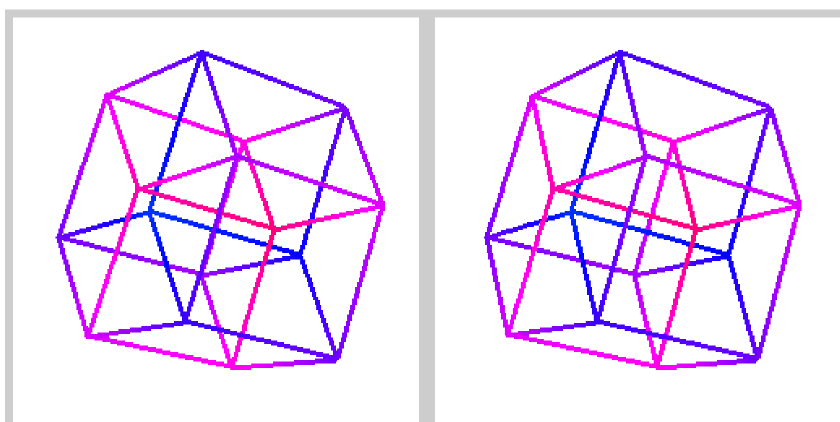

Figure 17: Tesseract projection (Academic.Ru, 2010).

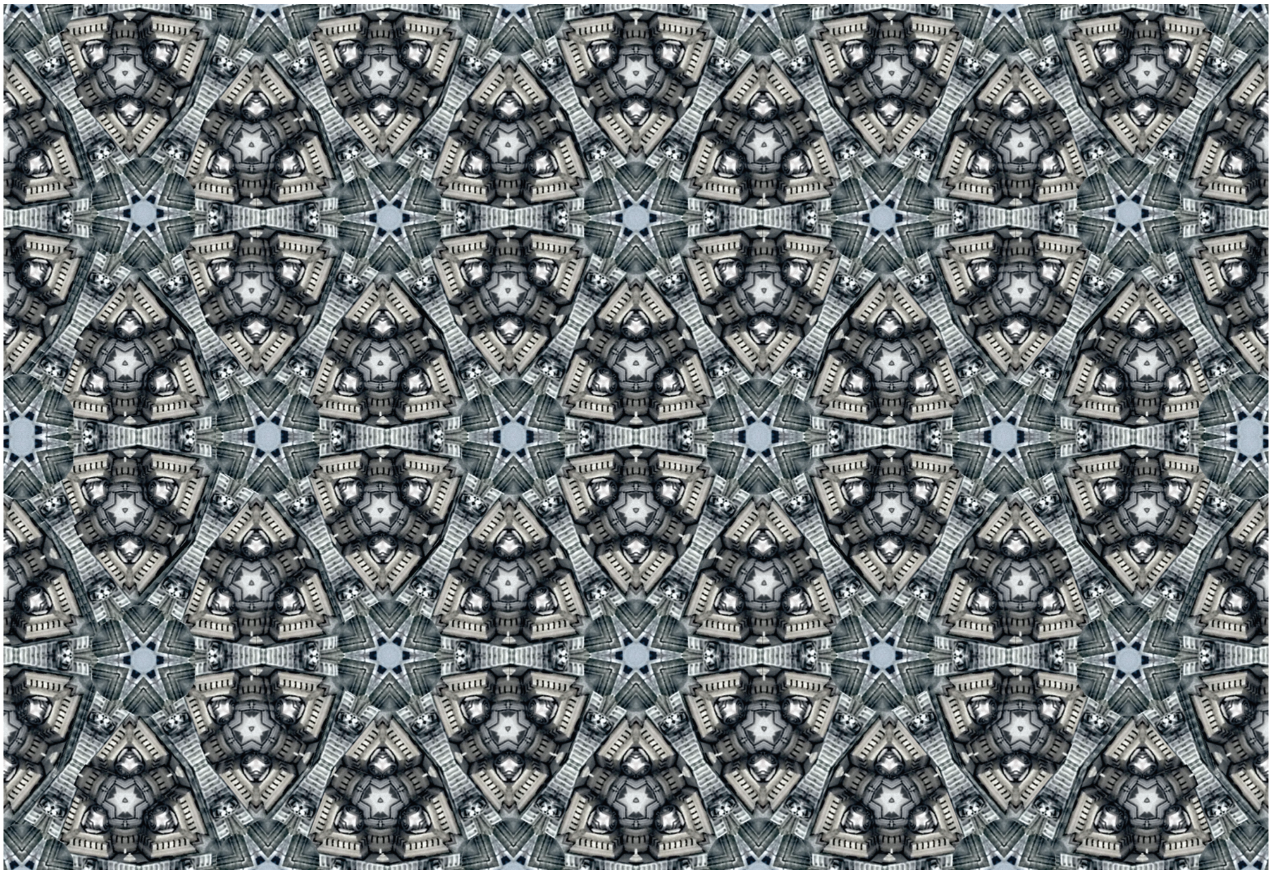

Figure 18: Mind-warping visualization (Norwood, 2018).

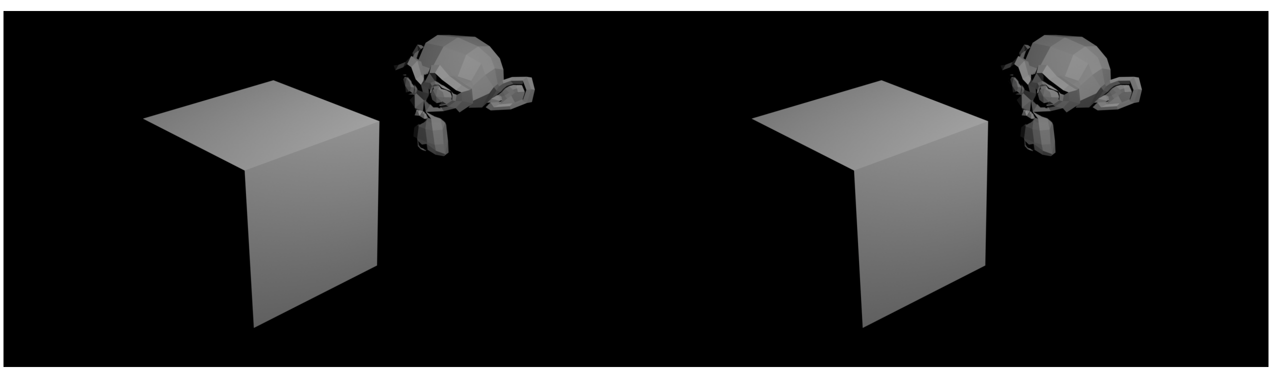

Figure 19: Blender's stereoscopic rendering capabilities.

6.4 Future Work and Applications

- A function that quickly converts stellar parallaxes to a z coordinate used in the program, allowing us to visualise globular clusters (an example) in 3D using the GAIA data.
- Cosmological redshifts could be converted to parallax to perceive the depths of galaxies.
- A class of functions to visualise molecules, which could be printed in textbooks.
- A class of functions to visualise geographical data, terrains etc.
- A class of functions to visualise complex graphs/networks.

- A projection of the celestial sphere could be made.

6.5 Bonus Application

Cross eyed 3D is a superpower to spot the differences!

Figure 20: Cross-eyed viewing enhances spot-the-difference puzzles (Credit: Muband at Japanese Wikipedia, CC).

6.6 Conclusion

We conclude that stereography is a powerful way of conveying information visually.

There is a lot of potential work in the area.

There are limitations on its adoption.

References

Academic.Ru (2010). Stereopara. https://dic.academic.ru/dic.nsf/ruwiki/228253. Accessed: 2025-07-29.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Morse, P. M. and Feshbach, H. (1953). *Methods of Theoretical Physics*, volume 1 and 2. McGraw-Hill Book Company, Inc.

Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928.

Munzner, T. (2014). *Visualization Analysis and Design*. A K Peters Visualization Series. A K Peters/CRC Press.

Norwood, M. (2018). 3d art. https://norwooddesigns.com/3D%20Art.html. Accessed: 2025-07-29.

Shambaugh, S. (2024). mpl_stereo: See your matplotlib plots in 3D by making stereograms and anaglyphs. https://github.com/scottshambaugh/mpl_stereo. Accessed: 2025-07-29.

Soule, J. P. (2025). Map of New York city (digital id: G90f336_113f). New York Public Library Digital Collections. Public Domain; accessed on 2025-04-16.

TRIAXES (2025). Parallel cross-eyed viewing method. Web Page. Accessed: 2025-04-15.