Real-Time Chromostereopsis
For Arbitrary Three-Dimensional Scenes

Introduction

Among the several ways to perceive stereoscopic ("3D") depth on a 2D screen, one is through diffraction by colours, known as **chromostereoscopy**. Combined with the correct lenses (ChromaDepth glasses) and colours displayed, a stereo image can be created.

Relative to other forms of stereoscopy, few chromostereoscopic tools exist and generating imagery that works in chromostereoscopy takes some work. It would prove useful to have a tool that allows the conversion of any 3D scene in realtime to a valid chromostereoscopic image.

To account for some degree of user/scene variance, one might also want a level of customisation with the creation of such an image, so more broadly we aim to answer the following question:

What are effective ways to display chromostereoscopic depth in real-time 3D scenes?

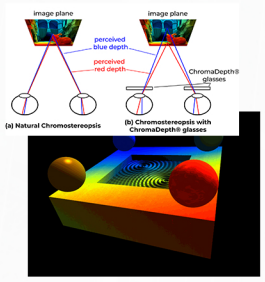

Fig. 1: A chromostereoscopic image. [1]

Creating Depth

Fig. 2: A depth buffer, visualised.

The first step to creating chromostereoscopic depth is to map depth to colour. In a scene that is already 3D, **depth has to be extracted per pixel.**

Luckily, modern computers can obtain this information fast enough for real-time rendering through what is known as the **depth buffer**. This is a greyscale image where each pixel's brightness maps to its depth relative to the scene's "camera", and is often used as a component for various effects in 3D rendering.

Adding Colour

As the name might suggest, chromostereoscopy relies on colours to map different depths, so the most common colour-depth mapping that works for chromostereoscopy is a **hue shift from red to blue.**

Achieving this can be done by "colouring in" the aforementioned depth map. The longest wavelengths (red) are perceived closest to the viewer, while the shortest (blue) are farthest. Through performing this all on the GPU in a shader, it can run fast enough to run in real-time even on smaller, mobile devices.

However, that alone is not the only colour mapping that works for chromostereoscopy, and many others exist, **such as one developed here at TU Delft** that is optimised for modern, trichromatic digital displays.

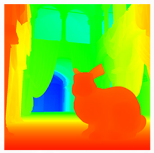

Fig. 3: A hue shift mapping.

Fig. 4: A trichromatic depth mapping.

Integrating Shading

Fig. 5: A scene rendered with PBR shading. [2]

The current state-of-the-art for real-time rendering is often considered to be shading that uses **physically-based rendering (PBR)**; a form of shading that obtains many of its lighting properties from real-world material data. This can be combined together with our colour-depth mapping to obtain "realistic" 3D scenes, still in real-time, that fully work with chromostereopsis.

This shading is combined in greyscale with the colour-depth mapping to not interfere with the colours that are necessary to produce the chromostereoscopic effect. **But what if we could still combine them?**

To further enhance the information provided in a chromostereoscopic image, we added the functionality to adjust the degree of which the scene's "**original colours**" blend together with the depth mapping.

The ratio of blending between the original colours to depth mapping colours will undoubtedly vary depending on the scene, but it is possible to display a degree of the scene's original colours, to add another degree of customisation and visual information to the image while maintaining the depth effect.

Fig. 6: A depth-colour mapping with shading applied and original colours.

Depth Precision

With colours assigned to depth values in a scene, there are only finite points on a colour map to assign depth to. Alternative depth mapping techniques may provide more depth detail in areas that have more use for it. Previous research has most often used **linear depth precision**, but people typically pay more attention to objects near to them.

Modern real-time graphics often make use of this inverse relation to depth in the scene, **prioritising nearby depth** far more than in the distance.

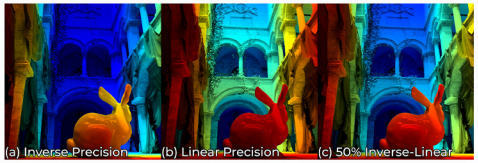

Fig. 7: A variety of depth precision forms, visualised.

Depending on the application, the focus of a 3D scene may prioritise different depths (e.g. geographical chromostereoscopy finds more use in linear mapping).

We provide a setting to **alternate between the two** forms on a percentage scale.

Depth in Motion

During camera motion, the colours created by the depth-colour map may sometimes change rapidly as an effect of the adjusting minimum and maximum depth of the view. This effect may potentially be intense for some users, so we provide a method to **slow down the speed of depth changes**, and prevent discomfort.

Per rendered frame, we store the minimum and maximum depth of the previous frame, and percentagewise blend the two values to ease transitions over time.

Additional Interfaces

Additional interfaces were created to assist in optimising available settings further towards a scene. In addition to the ability to load in existing gradients into the program, we added an interface to allow the creation of **arbitrary gradients**.

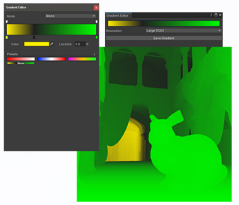

Fig. 8: Custom colour-depth mapping using the interface.

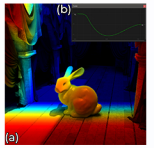

Fig. 9: Custom depth precision mapping.

Furthermore, to expand beyond linear and inverse depth precision scales, the ability to create **custom depth precision mappings** was added.

This system uses Bézier curves to define a mapping of choice, which is reflected in real-time.

Results

All techniques were evaluated subjectively through a small user study to measure preference and variance.

The depth-colour mapping has multiple valid colour maps with different appearances. The **Turbo** and **Jet** colour maps by Google and Matlab respectively [3;4], were most preferred.

Preference for modes of depth precision varied largely across the users, with most preferring some **combination of linear and inverse**.

Shading showed a **preference towards PBR shading** for its enhanced detail over no shading, or Lambertian shading used in previous work. When asked about preference towards blending in original colours, users in our scene preferred to leave this setting very low or off.

Fig. 10: Various colour maps for chromostereoscopy.

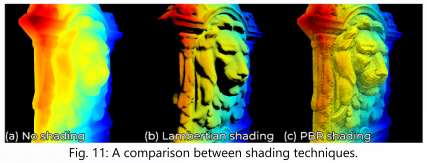

Fig. 11: A comparison between shading techniques.

During testing, the **depth motion smoothing was preferred by all**, despite no user experiencing discomfort or nausea with it disabled. The speed at which users preferred it adjusting exhibited only slight variance.

In conclusion, **all techniques presented showed a positive influence** on the resulting chromostereoscopic image by all users. Some options displayed a larger variance than others, indicating parameters that should be left as adjustable to user, or further research may prove useful into narrowing these down more.

References

[1] Michael Scroggins. Chromostereoscopy, 2022. https://michaelscroggins.wordpress.com/chromostereoscopy/
[2] Frank Meinl, Katica Putica, Cristiano Siqueria, Timothy Heath, Justin Prazen, Sebastian Herholz, Bruce Cherniak, and Anton Kaplanyan. Intel sample library, 2022 https://www.intel.com/content/www/us/en/developer/topic-technology/graphics-processing-research/samples.html
[3] Anton Mikhailov. Turbo, an improved rainbow colormap for visualization, 2019. https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html
[4] The MathWorks, Inc. jet, 2022. https://www.mathworks.com/help/matlab/ref/jet.html

Vladislav Gaidoukevitch - V.Y.Gaidoukevitch@student.tudelft.nl
Supervisor: Petr Kellnhofer; Responsible Professor: Elmar Eisemann

TU Delft Delft University of Technology