Simulating the ChromaDepth Effect for CMYK-based Print Media

Mike Verhoeff
Supervisors: Petr Kellnhofer, Elmar Eisemann
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Simulating the ChromaDepth Effect for CMYK-based Print Media

Mike Verhoeff
Supervisors: Petr Kellnhofer, Elmar Eisemann
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

Abstract

Chromastereoscopy makes use of special glasses to make its wearer see two slightly different images, one for each eye. With a properly created image this results in perceived depth. It does this by bending light dependent on its wavelength and in the opposite direction for each eye. When a ChromaDepth image is properly created most viewers perceive the image as having a 3D effect. However, since the glasses act on wavelengths and not on perceived colors, different display media like screens and printers could result in a different perception of the images through the glasses. To investigate this possible difference I measured reflectance spectra of printed colors, and created an algorithm to simulate the effect the glasses have on images made with mixes of these colors. Additionally I tried multiple visualization techniques to digitally show the effect including depth.

1 Introduction

The chromostereoscopic process has the ability to add apparent depth to a flat image while wearing ChromaDepth glasses[1]. In the past this has been used to add depth cues to scientific data, like height of terrain[2], marine data[3], and improving the ability to distinguish the veins in the vascular system[4]. Some examples these uses of ChromaDepth can be seen in Figure 1. The benefit that ChromaDepth has in these applications is that they can still be viewed as a regular image without the specialized glasses, as opposed to other stereo visualization techniques like anaglyphs[5]. Anaglyphs overlay a red and cyan image causing the parts of the image that are not at the depth of the paper to be displayed twice, each slightly offset from one another.

Since sharing images digitally is often easiest, whether that be scientific visualizations in papers or artworks, ChromaDepth images are often viewed on a screen. Subsequently most ChromaDepth art and research is done for screens. This leaves the application of ChromaDepth in physical media like print underdeveloped. To stimulate this development, I created an algorithm to simulate the ChromaDepth effect and its appearance without wearing the glasses.

The painting program code and the measured color samples can be found at: `https://github.com/MikeVerhoeff/ChromaPainter`

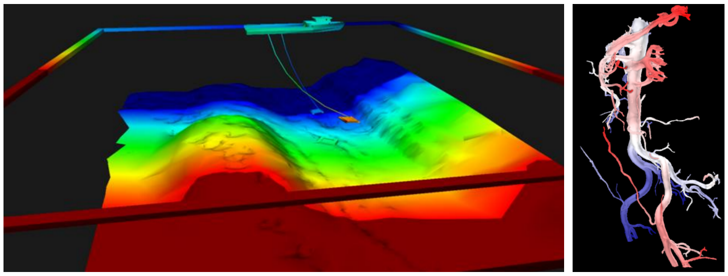


Figure 1: Left: Making positions of ships over the seabed more clear using ChromaDepth[3]. Right: Encoding positions of veins using a red, white, blue designed for screens[6].

One hurdle when it comes to simulating ChromaDepth glasses is that the glasses do not just bend the light based on the color humans perceive it as, but on the physical wavelength of the light. An object that reflects only yellow wavelengths of light will seem to have the same color as an object that reflects red and green light. But when wearing the glasses this difference can become visible at the edges of objects, since the red and green light are not shifted by the same amount and thus do not overlap completely anymore.

A second hurdle is that the result of simulating the glasses will still be a pair of flat images without depth. These two images together form a stereo image. This stereo image can be used in multiple different ways to create a visualization. First, stereo depth algorithms can be used to estimate the depth of each part in the image, however these algorithms often have difficulty when the lenses of the camera distort the image. Distorting the image by shifting colors is exactly what the ChromaDepth glasses do to get the depth effect. Second, using a Virtual Reality(VR) headset both color and depth can be shown. Normally when a scene is displayed in VR a stereo image is rendered from a 3D description of that scene. The description of an image however is in 2D, but by simulating the glasses a stereo image can be made that when applied to a plat plane in 3D space gives that plane an appearance of depth. The final visualization technique this paper describes is, creating an animation that has parallax motion to give an impression of the depth. Parallax motion is most clearly seen when traveling in a car, the trees by the side of the road move past quickly, while objects in the distance hardly move at all. Where in stereo vision the shift in perspective happens over space using the different positions of your eyes, with parallax motion the shift happens over time. The two images the glasses create can be enough to create apparent motion if the differences are small enough, but when the difference becomes larger intermediate images are needed to create motion. Since the changes the glasses create are small two images are enough to create a depth effect, but having more frames makes the animation look nicer.

Section 2 will give background information about spectra and colors. Section 3 will go over the problems I needed to solve to be able to simulate ChromaDepth, namely creating a spectral image, then running the simulation of the glasses on that image, and then visualizing the results. In this section I introduce how to simulate the glasses, and how the results of the simulation can be shown. In Section 4.1 I evaluate how well suited the YNSN model is for simulating my HP Photosmart 5520 ink-jet printer in order to get a spectral image. And show that while it does work for commercial printers it has difficulty simulating how an RGB image will print on a ink-jet for which no color profile is available. In Section 4.2 I verify that my simulation algorithm produces accurate results for printed images, but that the assumption that the glasses only diffract light to the first order can cause problems when using the same algorithm to simulate ChromaDepth on a screen. In Section 4.3 I go over the benefits and drawbacks of the visualization methods and conclude that parallax is currently the best method since it shows both color and depth without having to ware a VR headset. In Section 5 I will summarize the results of the evaluation and draw conclusions from it. Finally I will close with a section on responsible research, a discussion and a conclusion.

2 Background

The chromostereoscopic process - as first made practical by Richard Steenblik[1] - can add depth to an image by using color and stereo vision. He used two pairs of prisms, one in front of each eye. These prisms were placed to bend one wavelength of light more strongly than others while preserving focus. One configurations bent red light most strongly away from the nose making red objects appear closer than blue ones as seen in Figure 2.

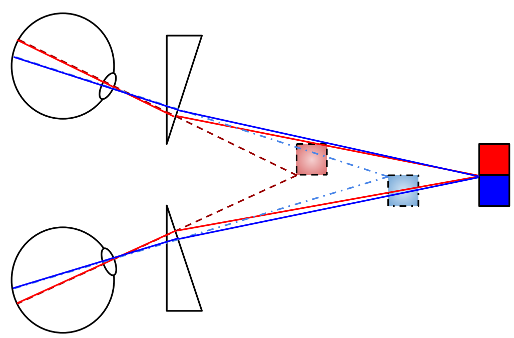


Figure 2: How the prisms and stereo vision work together to make colors appear at different depths.

While prisms work well for chromostereopsis they tend to be heavy and expensive making them impractical for wider use. To solve this problem a new sort of glasses were created by Chromatek[7] that were made of rows of tiny prisms etched into a transparent film. This makes the glasses work like a diffraction grating that refracts most of its light to the first order. At least in the case of 560nm light [8], when moving away from this wavelength more light goes to the other orders, For 633nm light only 75% goes to the first order [9]. An exact formula for this has not yet been found. How much each wavelength of light (color) gets bent is known and is given by this formula[10]:

$$m = d(\sin\alpha - \sin\beta)$$

Where m is the order of the diffraction, is the wavelength, d is the spacing of the repeating pattern on the diffraction grating and and are the incoming and outgoing angles of light. How a normal diffraction grating bends light can be seen in Figure 4.

In calculations involving light, like the one above, it is often represented using a wavelength. Most objects however do not reflect just one wavelength but partially reflect all wavelengths. The function that gives what fraction each wavelength is reflected is the reflectance spectrum $S(\lambda)$. The normal human eye has only 3 types of color receptors so using a function to represent color caries more data than needed. In 1931 the International Commission on Illumination (CIE) published the CIE XYZ standard [12], it specifies 3 values

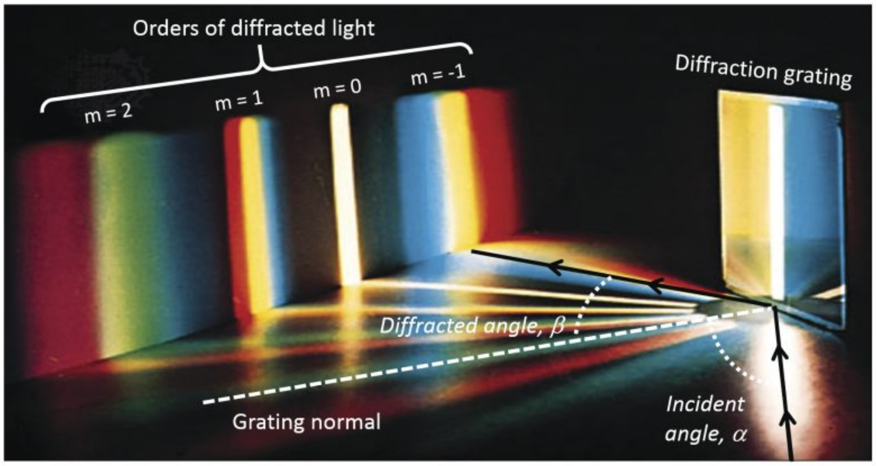


Figure 3: White light reflecting of a diffraction grating[11].

X, Y and Z to be able to capture all visible colors with values between 0 and 1. Since this XYZ color space can capture all colors it is often used to translate between different color spaces, like sRGB used for screens.

$$X = \frac{1}{N}\int_\lambda \bar{x}(\lambda)S(\lambda)I(\lambda)d\lambda$$

$$Y = \frac{1}{N}\int_\lambda \bar{y}(\lambda)S(\lambda)I(\lambda)d\lambda$$

$$Z = \frac{1}{N}\int_\lambda \bar{z}(\lambda)S(\lambda)I(\lambda)d\lambda$$

$$N = \int_\lambda \bar{y}(\lambda)I(\lambda)d\lambda$$

Formulas for the XYZ values bases on a reflectance spectrum $S(\lambda)$, light source $I(\lambda)$ and standard observer function $\bar{x}$, $\bar{y}$, $\bar{z}$.

Only using measured reflectance spectra would limit the amount of colors available for images. To allow for the use of more colors I make use of two models that simulate mixing of colors. One the Yule-Nielsen modified Spectral Neugebauer (YNSN) model[13]. This model is based of the chance that a photon of light hits a peace of paper with a particular color ink on it. Two a simplified version of the Kubelka-Munk model. The full Kubelka-Munk model[14] models a colored material as a stack of layers each absorbing (k) and reflecting (s) some of the light and letting the rest pass though. The simplified version reduces the two values k and s to a ratio between the two k/s, allowing it to be calculated from a single reflectance value. This simplification is appropriate when a ink is used that dyes the paper, like in the case of ink-jet printers[15].

3 Simulating ChromaDepth

Designing art digitally allows artists to more quickly create their artwork and experiment more easily since changes can easily be undone. However the digital design of ChromaDepth images have a significant drawback. Namely that even when the color on screen and in real life look the same, the light that creates the color might be composed of different wavelengths. Since the ChromaDepth glasses act upon these wavelengths of light, the effect on a screen and on paper can look different.

Thus before I can tackle the problem of creating a simulation I first need to be able to create a spectral image. How I create spectral images I discuss in Section 3.1. After that in Section 3.2 I go over how to simulate what effect the ChromaDepth glasses have on the image. Finally in Section 3.3 I will propose techniques to show a ChromaDepth image digitally and without glasses while still giving an idea of depth.

3.1 Spectral Images

Commonly used painting software like Photoshop, Paint, and GIMP represent their images using RGB values. However to be able to simulate the ChromaDepth glasses a reflectance or emittance spectrum should be available for each pixel. To be able to work with spectral images I created my own painting program. The reflectance spectra used in this painting program were measured using an x-rite i1 pro. This spectrometer produces 36 floating point values with a resolution of 10 nm. Higher end spectrometers have a higher resolution and thus more values to store. To limit the amount of memory needed I only store the spectrum once for the entire image, and reference the spectrum for each pixel. Only allowing one color per pixel would limit the user of the painting program to the colors they measured and make it difficult to create smooth gradients, since those need a lot of colors. To solve this problem I simulated a halftone ink-jet printer to get intermediate colors. A halftone printer creates colors by putting dots of ink on the paper. Larger dots for a color makes that color more prominent. The Yule-Nielsen modified Spectral Neugebauer (YNSN) model[13] estimates what area of the paper is covered by each dot and how much they overlap. Then using these areas combines the reflectance spectra of the colors that cover that area.

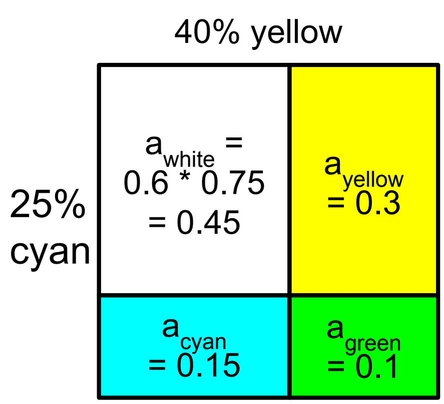


Figure 4: The areas that the YNSN model used.

Number of the color	binary	ink combination	Color
0	00	None	White (the paper)
1	01	only Cyan	Cyan
2	10	only Yellow	Yellow
3	11	Yellow and Cyan	Green (Mix of the 2)

Table 1: Assigning numbers to colored areas in the YNSN model, when only yellow and cyan ink are used.

$$R(\lambda) = \left(\sum_{n=0}^{2^k} a_k R_k^{1/n}(\lambda) \right)^n$$

Yule-Nielsen modified Spectral Neugebauer (YNSN) equation: a_k is the area covered by color k. $R_k(\lambda)$ is the reflectance spectrum of color k. n is accounts for various physical phenomenon and has to be found experimentally[13]

To keep track of all the different colors and areas I assign a number to each. Each bit in the number corresponds to one of the inks the printer is using. If a bit in the number is 0 then that ink is not used to create the numbered color, if the bit is 1 it is used. The results of this assignment for 2 dyes can be seen in Table 1. The mixes of these colors can either be measured or again be simulated. Since my printer does the RGB to CMYK conversion by itself I could not print the mixes I needed, so I simulated it. For the simulation I used A simplified version of the Kubelka-Munk model that assumes that the dyes do not reflect light, only absorb or transmit it[15].

$$(K/S)_{\lambda,coloronpaper} = \frac{(1 - R_\lambda)^2}{2R_\lambda}$$

$$(k/s)_{\lambda,color} = (K/S)_{\lambda,coloronpaper} - (K/S)_{\lambda,paper}$$

$$(K/S)_{\lambda,mixonpaper} = (K/S)_{\lambda,paper} + \sum_i (k/s)_{\lambda,i}$$

$$\hat{R}_\lambda = 1 + (K/S)_{\lambda,mix} - \sqrt{(K/S)^2_{\lambda,mix} + 2(K/S)_{\lambda,mix}}$$

R_λ is the measured reflectance spectrum, $\hat{R}_\lambda$ is the simulated reflectance spectrum of the mix.

3.2 Simulating the Glasses

Since the glasses create the depth effect by shifting colors, I based my simulation algorithm on these shifts. To calculate the shift I made two assumptions about the glasses. First, they act as a diffraction grating that only refracts light to the first order. While some light does go to the higher orders of diffraction this does not large differences in my testing setup. Second, that light exits the glasses perpendicular to the glasses. This is akin to an orthographic projection, instead of a normal perspective projection.

To get a formula for the shift I start with the formula for a diffraction grating.

$$m\lambda = g(\sin\alpha - \sin\beta)$$

Than I apply the assumptions that it is an orthographic projection ($\beta = 0$) and that it only refracts to the first order ($m = 1$).

$$\lambda = g\sin\alpha$$

$$\alpha = \arcsin\frac{\lambda}{g}$$

Using this alpha I can calculate a shift as outlined in Figure 5.

$$shift = \tan(\arcsin\frac{\lambda}{g}) * distance$$

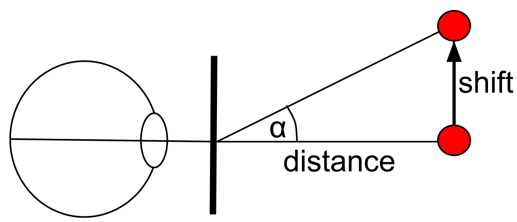


Figure 5: The glasses create a shift by bending the light.

The simulation algorithm works on a spectral image, and calculates what spectral image an eye will see. To do this it also needs a distance the image is viewed at and the size of the pixels, since the calculations for how much a wavelength shifts by are in meters. The shift of each wavelength only depends on the wavelength itself, the spacing in the diffraction grating and the distance to the image one shift can be calculated for each wavelength since the other variables do not change while running the algorithm. After converting this shift from meters to pixels, each intensity of a given wavelength in the spectral image can be shifted by the appropriate amount of pixels. The direction that it is shifted in determines for which eye it is. This direction of shifting can be incorporated into the formula by adding a variable - *eye* - into the formula for the angle.

$$\alpha = \arcsin\frac{\lambda * eye}{g}$$

When *eye* is 1 the image is calculated for the left eye, and for -1 it is calculated for the right eye. The image does need to be extend so there is space for the pixels that get shifted out of the range of the original image. To be able to display this spectral image it is converted to an RGB image. To do this it is first converted to the CIE XYZ color space before being converted to the sRGB colorspace.

3.3 Visualization

The ChromaDepth glasses present two different images, one to each eye. In this way it makes use of stereo vision to create the depth effect. However when these images are simply displayed next to each other on a screen the human visual system does not merge them together into one image with depth. Other methods are thus needed to display what effect the glasses have when it comes to the depth effect that they create. The methods for adding depth discussed in this section are: creating relative movement/parallax, displaying the image in Virtual Reality (VR) headset, and generating a depth map.

Parallax

To create an animation that has relative motion it is possible to use just the two images that each eye sees and switch between them, this however does not look nice and can stop working when the differences become to large. To create an animation that has smoother movement, more frames in between should be calculated. For this I used the *eye* variable with values between -1 and one 1. This results in a smooth transition between the left and right image. However the glasses shift all the wavelengths in the same direction and the relative shift between the visible wavelengths is quite small. This causes the image to shift from left to right so much that I did not perceive any depth from relative motion. To be able to see the relative motion clearly I fix one wavelength in place by shifting the image back the same amount as that wavelength was shifted by the simulation. I perceived the most clear depth when a blue wavelength like 530 nm was fixed in place. This causes the background of the image to stay in place while the foreground moves.

Virtual Reality

When the user is wearing a VR headset the computer can display an image for each eye and thus create depth. To show the ChromaDepth effect in VR, I textured a plane so that the left eye sees the simulated image for the left eye, and the right eye sees the image for the right eye. In Unity this can be done by creating a shader that is aware of which eye it is rendering for. However using the raw output of the simulation makes the image appear closer than it should be since all the wavelengths are shifted towards the nose. To make green appear at the depth of the plane the simulated images have to be shifted back so that green stays in the same place between the two images. Doing this makes red still appear in front and pushes blue to appear behind the plane, like when looking at a chromadepth image on paper.

Depth Map

Stereo depth estimation algorithms are often used to create a depth map from two camera feeds. Since the simulation can create two images a stereo depth algorithm can be used with it. The algorithm I tried is Stereo Block matching as implemented in OpenCV. All the parameters that were not required were left at their default values. The numDisparities parameter changes between 0, 16, 32 and 64 to find the best result. Likewise the block size was set as an odd number between 7 and 21.

4 Evaluation

Before evaluating the results of simulating the glasses I will evaluate the accuracy of the color mixing. This is to see if discrepancies between the results of simulation and reality are caused by inaccurate color mixing or if they are errors in the simulation. After having evaluated the accuracy of both the color mixing and simulation of the glasses I will evaluate the different visualization methods.

4.1 Color Mixing

To evaluate the accuracy of the color mixing I did four experiments. In all these experiments I use a set of color samples measured using an x-rite i1 pro. A small amount of these samples are used by the mixing algorithm as color bases, and all samples are used to tune the n parameter of the YNSN model. To then assess the accuracy of the color mixing simulation I simulate the mixing of the sampled colors from the base colors and calculate an average DeltaE 1976 between the simulation and the sample. When the DeltaE is less than one a typical human can not see a difference; between 2 and 10 while the difference is clearly visible, the colors are still close; at 100 the colors are the exact opposite [16].

The first experiment is performed using 39 samples from a commercial printing color reference card that used Offset Lithography. For which CMYK color values are available. The mixing simulation only used 5 of these samples namely pure cyan, magenta, yellow and black, plus the paper without ink. The mixed colors YNSN needs (Neugebauer primaries) are calculated using the Kubelka-Munk model. This gives an average DeltaE of 4.16 at n=2.17. In the second experiment the Neugebauer primaries that are mixes of two or four base colors are replaced with measured values. The mixes of 3 colors were not available on the color reference card. This improved the DeltaE to 2.21 at an n of 2.11. These 2 results indicate that YNSN gives satisfactory results in my test setup. The results also line up with previous evaluations of YNSN where an average DeltaE94 of between 0.5 and 3 was found depending on the specific ink, paper and printer, when only using measured Neugebauer primaries [17]. However, the use of the Kubelka-Munk model to calculate the Neugebauer primaries harms the quality significantly since in experiment 1 the DeltaE is double that of experiment 2 where Kubelka-Munk is not used; this model is not suitable for offset lithography.

The third and fourth experiment use a different set of 50 color samples created using an HP Photosmart 5520 inkjet printer. The printed colors were specified using RGB values since the printer does the RGB to CMYK conversion internally. In the third experiment RGB to CMYK conversion was done using the SWOP2006 color profile, in the fourth experiment the conversion was done by using the following formulas.

$$k = 1 - max(r, g, b)$$

$$c = 1 - r - k$$

$$m = 1 - g - k$$

$$y = 1 - b - k$$

When using the color profile the average DeltaE is 11.91 with n=2.8. And when using the formulas for conversion the average DeltaE is 8.52 with n=5.22. These values are high when compared to the first two experiments. Since in the fourth experiment CMYK was assumed to be the opposite of RGB and the DeltaE was double the worst result of the first two experiments, you can see that CMYK and RGB are not exact opposites. This can also be seen when looking at a RGB blue and a CMYK mix of 100% cyan and magenta. The mixed color is more purple than the RGB blue. To create a blue using CMYK less magenta is used than cyan. This can clearly be seen though the ChromaDepth glasses when blue is printed next to a magenta background since the blue shifts more than the magenta and reveals a light magenta line. This can be seen in Appendix A.

To conclude, while YNSN can create a good mapping between CMYK color values and the actually printed color when samples of all Neugebauer primaries are used. The mapping from RGB to a color, as printed by a HP Photosmart 5520, using YNSN preforms less well. This is most likely because no color profile is available to do the RGB to CMYK conversion the same way the printer does. This means that YNSN is not a good model to simulate printing RGB images on a printer that does not have a color profile available.

4.2 Simulating the Glasses

When using my printer the color simulation has a large error; when comparing the simulation to real life I will only use measured color samples. To compare the simulation to reality I placed the right lenses of the ChromaDepth glasses in front of my phone camera and positioned it half a meter above the printed image with a bright 5000 Kelvin led light just below and to the side of the phone camera. A 3D model of the setup can be seen in Figure 6. The simulation used a distance of 0.4 m and a pixel size of 0.26E-3 m (CSS standard).

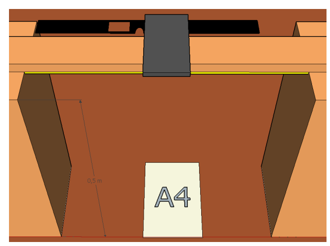


Figure 6: 3D model of the measurement setup.

First I will compare the primary and secondary colors on a black and white background as can be seen in Figure 7. Starting on the left with the black vertical line I see two differences namely that the black and white in the photo are darker and that the chromatic aberrations are less bright. Since both the solid blocks of black and white are less bright and the measured samples were produced by the same printer that printed the image, the difference is likely because of the camera and the lighting. Apart from the brightness the chromatic aberrations look similar, on the left there is a cyan and a blue line of which the blue one is smaller and on the right there is a red and yellow line of which the red one is smaller. Since the brightness difference is likely caused by the camera and lamp, I find that in this case the simulation is accurate. For the rest of the lines of color similar observations can be made and again apart from brightness the simulation is accurate.

Another observation that can be made is which direction the colors shift when on a white background compared to a black background. Here again the simulation and reality line up, with cyan and blue shifting to the left, yellow magenta and red shifting to the right and green staying in the middle. Similar observations can be made when using the colors of the vertical stripes as background colors. For an image that also includes these see appendix A. There is one aspect of the glasses that the simulation does not take into account, namely that some light gets refracted to a higher order of diffraction this can become visible when the background is dark and the color is bright like in Figure 8. For a bright screen this effect

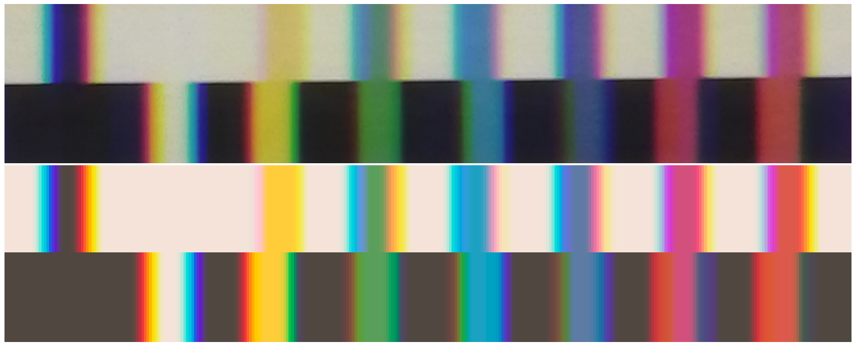


Figure 7: On top a photograph through the ChromaDepth glasses. On the bottom a simulation of that same image.

is very clear, but on paper this effect is much less noticeable.

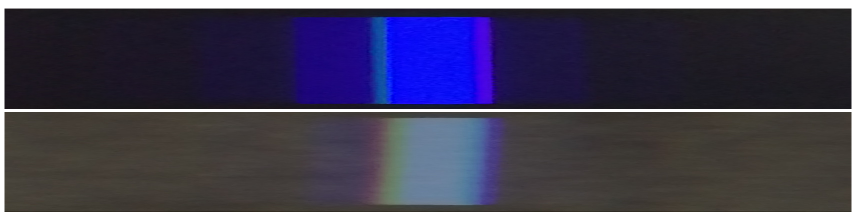


Figure 8: On a screen(top) light gets visibly refracted to others, For paper(bottom) this also occurs but less strong.

The simulation algorithm has been written to run on a single thread. Unfortunately this makes it quite slow needing about one second for an image of 300 by 300 pixels. However when comparing it to printing an image which takes about 12 seconds on my HP Photosmart 5520 it still compares favorably. The runtime of the algorithm scales linearly with the number of pixels as can be seen in Figure 9. To improve the runtime of the algorithm it can be parallelized since all calculations can be done per pixel this should not give many problems.

4.3 Visualization

Up till now the simulation has produced an image but this image does not have depth. In the following section I will evaluate how well the different visualization techniques work. Creating an image with relative movement produces good results with more complex images that include color gradients and other depth cues like shadows. One can However experience problems with images that are too simple and only include a few far apart areas of color like in Figure 10.1. In these cases while the movement is visible the other objects are to far away to see relative movement between then and thus no depth is observed.

In Virtual Reality I displayed a plane with the appropriate texture for each eye about 3 meters away and 2 meters high. Next to it I put a plane of the same size but with the image as viewed without the chromadepth glasses. The plane that had the different textures for each eye clearly had depth while the plane with only one texture had shown no depth.

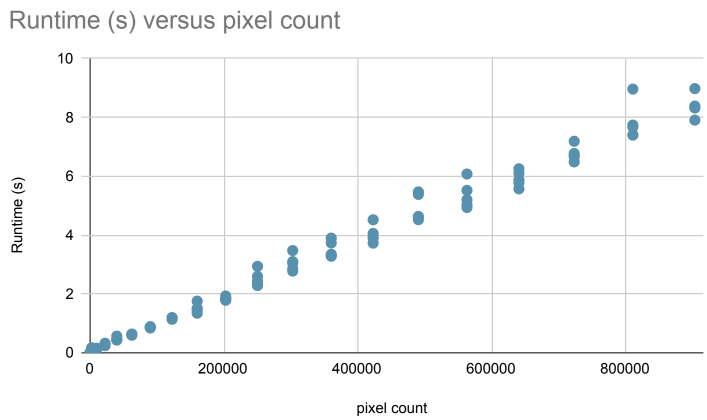


Figure 9: The single threaded runtime of the simulation algorithm for images of different sizes on an Intel Core i7 3630QM (Laptop processor from 2013).

The main drawback of VR is that it requires special hardware that is not ubiquitous, and putting on the headset and taking it off can be a nuisance.

Creating a depth map using stereo block matching to create a disparity map had mixed results as can be seen in Figure 10. It worked well on images that have clearly distinct objects that were either small or had a lot of texture that made parts darker but did not change the color. It performed badly when there was little texture. One area where it performed better than the simpler technique of using the color of the pixel as the depth was when the background made it look like different colors were in front. In cases like these the depth map correctly predicted the ordering of the colors.

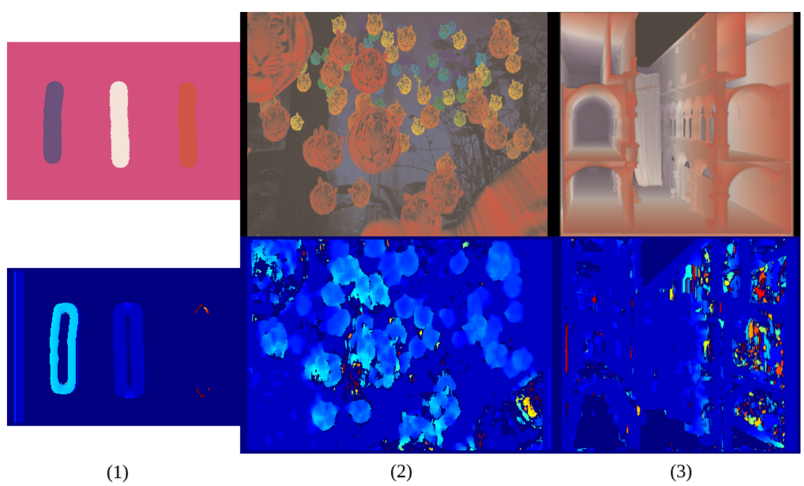


Figure 10: A ChromaDepth image(top) and a disparity map(bottom) using the jet color map for it created by stereo matching.

5 Results

The depth effect that the ChromaDepth glasses give can be simulated and displayed digitally given a spectral image. If no spectral image is available it can be created by simulating a printer. Using the YNSN model can give good results given a CMYK image and a professional printer. However when only a RGB image is given and the printer has no color profile available - to emulate its RGB to CMYK conversion externally - the results are usable but less than optimal.

To simulate the effect of the glasses, the reflectance values for each wavelength are shifted by the same amount a diffraction grating would shift the first order diffraction of that wavelength at a given distance. While the glasses do refract light to higher orders, when looking at printed images this is hardly noticeable. However when looking at a screen this does become clearly visible. Finding a formula for the degree to which this happens can be done in future work, and will allow for the improvement of the simulation.

Of the three methods used to show the depth effect without the glasses, parallax is the method with the fewest drawbacks. It can give a good idea of the depth without needing to wear a VR headset. It also shows color and works well in most circumstances. Unlike stereo block matching to create a depth map, which only works well for simple images. However, creating a depth map might work better with better stereo depth estimation algorithms.

6 Responsible Research

Chromasterioescopy has been used for both medical and scientific data visualization. This paper has found that when images are printed some colors can display more distortion than when viewed on a screen. On the printer that was tested, a HP Photosmart 5520 this had the most significant effect on blue. In the case of scientific data this could cause readers to misinterpret the data or make the results less clear. In the case of medical visualization misinterpreting the data could be dangerous for the patient. To prevent these problems from occurring I recommend specifying that an image should only be viewed digitally if it has not been verified to give the correct effect when printed using a specific printer. This verification has to be done by printing it and not by simulating it using my simulation since it has problems with the color conversion without a color profile.

7 Discussion

The painting application developed for this project was created in order to have images to run the simulation on and allow for quick experimentation. The interface was however not designed to make the creation of nice looking artworks easier. Creating a user interface on top of these simulations to make designing a ChromaDepth image better could be done in a later study.

The depth estimation, while far from perfect, showed promising results. If a good estimation of perceived depth can be achieved - by using better or more suited algorithms - then the depth map could be used to better evaluate the coloring programmatically. This might allow for a new way to color ChromaDepth images where instead of using the of depth of one pixel as basis for the color of that pixel all the pixels in the image are considered. By finding colors that make the actual depth map and the estimated depth map the most similar.

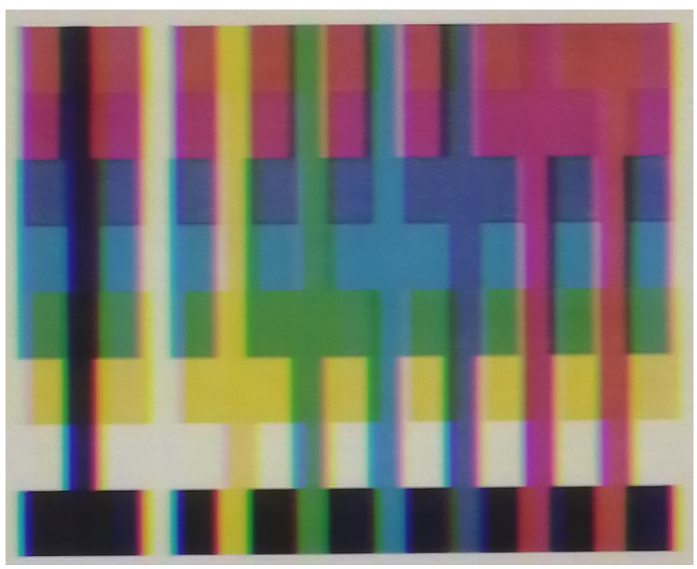
To display ChomaDepth in VR I textured a plane using a stereo image. This made the plane appear to have depth just like a ChromaDepth images in reality, even though the sterio image was static. This same idea could also be used in general usage of VR to give surfaces extra depth without using extra geometry, like the way normal maps are used to made a surface appear less smooth. This does however require an extra texture, and thus might not be worth the reduction in geometry.

8 Conclusion

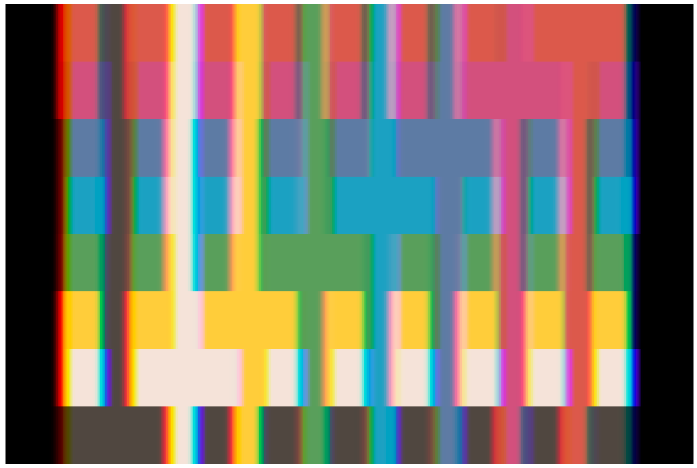
I presented a way to simulate the ChromaDepth glasses for looking at print media, and created a painting program to create digital representation of these print media using reflectance spectra of the inks used. The simulation allows for the quicker evaluation of printed ChromaDepth images than having to print the image. And thus allow for quicker exploration and creation of images using the ChromaDepth effect on paper. Having a good way to simulate the effect will also allow for more objective evaluation of the effect in future papers.

A Real and simulated image of colors on different backgrounds

Photograph of printed image through ChromaDepth glasses



Simulation of printing and ChromaDepth glasses



References

[1] R. A. Steenblik, “The chromostereoscopic process: A novel single image stereoscopic process,” in *True three-dimensional imaging techniques & display technologies*, vol. 761. International Society for Optics and Photonics, 1987, pp. 27–34.

[2] T. Toutin, “Qualitative aspects of chromo-stereoscopy for depth perception,” *Photogrammetric Engineering and Remote Sensing*, vol. 63, no. 2, pp. 193–204, 1997.

[3] I. Abdel Hamid, “Chromo-stereoscopic visualisation for dynamic marine operations,” 2012.

[4] B. Behrendt, P. Berg, B. Preim, and S. Saalfeld, “Combining Pseudo Chroma Depth Enhancement and Parameter Mapping for Vascular Surface Models,” in *Eurographics Workshop on Visual Computing for Biology and Medicine*, S. Bruckner, A. Hennemuth, B. Kainz, I. Hotz, D. Merhof, and C. Rieder, Eds. The Eurographics Association, 2017.

[5] M. Bailey and D. Clark, “Using chromadepth to obtain inexpensive single-image stereovision for scientific visualization,” *Journal of Graphics Tools*, vol. 3, no. 3, pp. 1–9, 1998.

[6] L. Schemali and E. Eisemann, “Chromostereoscopic rendering for trichromatic displays,” in *Proceedings of Expressive 2014 (ACM Symposium on Non-Photorealistic Animation and Rendering)*. New York, NY, USA: ACM, August 2014, pp. 57–62, doi>10.1145/2630397.2630398. [Online]. Available: http://graphics.tudelft.nl/Publications-new/2014/SE14a

[7] Chromadepth3D, “Home,” 2015. [Online]. Available: https://chromatek.com/

[8] C. Ucke, “3-d vision with chromadepth glasses,” 01 1999.

[9] J. Sicking, T. Toepker, and G. Wojtkiewicz, “More than meets the eye,” *The Physics Teacher*, vol. 33, no. 7, pp. 446–448, 1995. [Online]. Available: https://doi.org/10.1119/1.2344263

[10] Dec 2018. [Online]. Available: https://www.shimadzu.com/opt/guide/diffraction/02.html

[11] [Online]. Available: https://www.newport.com/n/diffraction-grating-physics

[12] T. Smith and J. Guild, "The c.i.e. colorimetric standards and their use," *Transactions of the Optical Society*, vol. 33, no. 3, pp. 73–134, jan 1931. [Online]. Available: https://doi.org/10.1088/1475-4878/33/3/301

[13] M. Hebert, D. Nebouy, and S. Mazauric, "Color and spectral mixings in printed surfaces," vol. 9016, 03 2015.

[14] "An article on optics of paint layers."

[15] L. Taplin, "Spectral modeling of a six-color inkjet printer," 2001.

[16] Z. Schuessler, "Delta e 101," 2014. [Online]. Available: http://zschuessler.github.io/DeltaE/learn/

[17] M. Hebert and R. D. Hersch, "Review of spectral reflectance models for halftone prints: Principles, calibration, and prediction accuracy," *Color Research & Application*, vol. 40, no. 4, pp. 383–397, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/col.21907