Underwater Colorimetry - Course Project

Due Date: May the 4th:-)

Contents

1 Project Overview

Goals of the project:

- **Part A:** Using the forward (physical) image formation model to simulate color chart images in various underwater imaging scenarios.
- **Part B:** Using the backward (wideband) image formation model to remove water from real-world underwater images and convert them to a standard `RGB` color space.

Deliverables: You may work in groups, however, each student will submit an individually written project report.

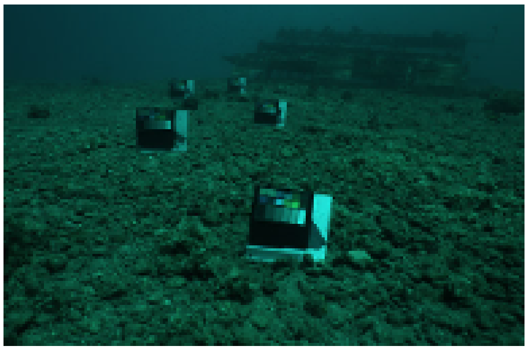
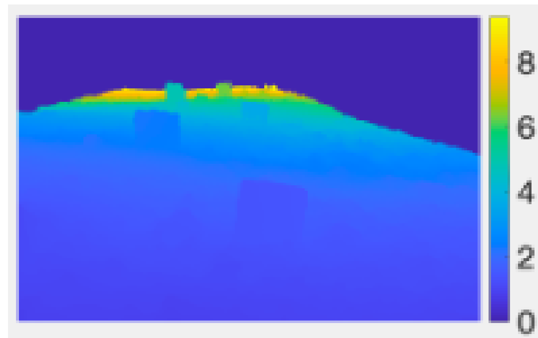
- The deadline to submit your report is May 4, 2025 @ 23:59.
- The report format is PDF.
- Name your file `lastname_firstname_UWCreport.pdf`.
- Email to: **uwcolorimetry@gmail.com**.

Grading: The project is 80% of your total grade. Grading will not only be based on your results but also on the effort you put in, and whether you displayed an understanding of the concepts. So, please take care to explain/discuss your findings.

- `ChatGPT` is helpful with coding and feel free to use it for that purpose, but it really does not understand underwater vision. If you will ask `ChatGPT` to do your entire project, we will probably catch that, and you will get a grade of 0.
- And if you do it and we don't catch it—who did you cheat, really? Us, or yourself?

Real-world data assignments: In Part B of the project, each student will analyze two real-world images. One image comes from a clear body of water (Eilat), and the other comes from a turbid body of water (Florida). The images are in the course github repository.

Student	Eilat Image	Florida Image
Amit	1	1
Shai	2	2
Nir	3	3
David	4	4
Itamar	5	5
Barak	6	6
Nitzan	7	7

RGB Image (png format)	D Image scaled in meters (tiff format)
	

Some notes about the real-world data:

1. We provide you RAW images for each dataset. Your first task is to convert them to linear .tiffs, (or if needed for speed, to linear .pngs).

2. We also provide you depth maps ("D" images) in .tiff format. These were made in Agisoft MetashapePro. When Agisoft exports depth maps, it puts a value of zero (0) to pixels for which no depth information was calculated (e.g., water pixels). In our nomenclature, a distance value of 0 denotes the location of the camera and theoretically it is impossible to capture an object at $z = 0$. For your project, you don't need to access water pixels, just be aware 0 is not a valid value. Replace 0 values with NaNs.

3. Often, the furthest distance values in a D image are inaccurate. If you notice your data behaving weirdly at very large distances, filter out those D values.

4. These are real-world data and they are not perfect. For some datasets, the D image values may not exist for some of the far-away charts. All you need to do these exercises are to have 2 color charts with D information. Therefore, even if you do not have D for ALL color charts, you can still do everything we ask.

5. The Eilat dataset was collected with a compact Sony (RX100 MKVII), and the Florida dataset was collected with a GoPro 11. You have the spectral response curves of both cameras in the course github.

Instructions

2 Part A - Simulation

2.1 RECAP: Forward image formation model:

In this part we will use the forward image formation model (the one that is a function of wavelength) to simulate the appearance of color charts underwater, under different optical conditions. As we saw in the lectures, the integral form of the underwater image formation model goes as follows:

$$I_c = \overbrace{\int_{\lambda_1}^{\lambda_2} \left[\rho(\lambda)E_0(\lambda)e^{-K_d(\lambda)D} \cdot e^{-c(\lambda)z}\right] \cdot S_c(\lambda)d\lambda}^{Direct\ signal\ (D_c)} + \underbrace{\int_{\lambda_1}^{\lambda_2} \left[\frac{b(\lambda)}{c(\lambda)}E_0(\lambda)e^{-K_d(\lambda)D}(1 - e^{-c(\lambda)z})\right] \cdot S_c(\lambda)d\lambda}_{Backscatter(B_c)}$$

Where:

$$\begin{aligned}
&\lambda - Wavelength \\
&S_c(\lambda) - Sensor\ spectral\ response \\
&\rho(\lambda) - Target\ reflectance \\
&E_0(\lambda) - Diffuse\ downwelling\ light\ at\ the\ surface \\
&b(\lambda) - Total\ beam\ scattering\ coefficient \\
&c(\lambda) - Total\ beam\ attenuation\ coefficient \\
&K_d(\lambda) - Diffuse\ downwelling\ attenuation\ coefficient\ [m^{-1}] \\
&z - Distance\ along\ line\text{-}of\text{-}sight\ between\ camera\ and\ the\ scene \\
&D - Depth\ from\ the\ surface\ [m]
\end{aligned}$$

2.2 Exercise 1: Simulate a color chart underwater

1. Until now, you used a Macbeth ColorChecker in your simulations. However, Macbeth charts are not waterproof, so in real-world applications, we use color charts from a company called DGK. Whereas Macbeth charts have 24 patches, DGK charts have 18 patches. We provide you the reflectances of patches of the DGK chart in your course github repository.

2. Use illuminant D-65 as the diffuse downwelling light at the surface.

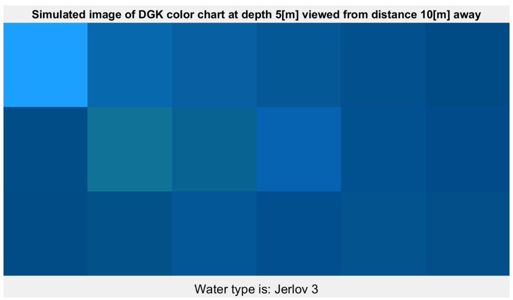
3. Use the Canon_1Ds-Mk-II camera as your sensor (from Lab 1).

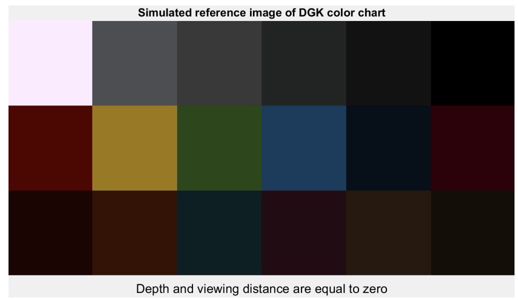
4. Write the functions **simulateBackscatter** and **simulateDirectSignal**.

5. Write the function **simulateImage**, which simply returns $I_c = D_c + B_c$.

6. Use your functions to simulate the appearance of the DGK chart in clear air (hint: no B_c). What is the value of the white patch? White balance the image with that value (and record it, because you will use it soon), so that the value of the white patch in the "in-air" simulation becomes [1 1 1].

7. Simulate the DGK color chart under several conditions. Choose **AT LEAST** 3 different combinations of: [water type - water depth - scene depth], e.g., Jerlov Type 3, 5m depth, 10m along the line-of-sight (see example below).





8. Make sure you white balance all your simulated images with the white value from the "in-air" chart. This way, you can do a visual comparison.

What to include in your report

1. Simulated DGK color chart in air.

2. All the other color chart images you simulated. Label them properly so we know what we are looking at.

3. Include a plot of the physical coefficients $K_d(\lambda)$, $b(\lambda)$, and $c(\lambda)$ associated with each simulated combination.

Discuss :

1. How do the physical coefficients, especially $K_d(\lambda)$, explain the resulting patch colors in the images?

2. How do depth and viewing distance affect the color of the patches? Can you tell which affects the most? Is that effect always the same across the different water types you simulated?

2.3 Exercise 2: Direct signal (D_c) and Backscatter (B_c)

1. Use the same 3 combinations from Exercise 1, and visualize just the direct signal (D_c) for the color chart.

2. Use the same 3 combinations from Exercise 1, and visualize just the backscattered signal (B_c) for the color chart.

What to include in your report:

1. Include visualizations for the D_c and B_c (both in color chart layout, so 3 x 6) for each of your 3 scenarios. Label them properly.

Discuss :

1. What do you notice about D_c? Are all colors affected in the same way?

2. What do you notice about B_c? Are all colors affected in the same way?

3. In the D_c and B_c signals you observe, what do you think is the effect of water depth D? What about viewing distance z?

2.4 Exercise 3: Varying optical properties and viewing distance

Now let's compare the appearance of 5 color charts, each at a different distance, imaged in two different water bodies. This example parallels that you will do in Part B.

1. Use water types 2 and 8 from the Jerlov dataset.

2. For water type 2, assume water depth $D = 15m$, and simulate the I_c, D_c, and B_c for 5 color charts at distances $z = 2, 4, 6, 8, 10, 20, 30$ meters.

3. For water type 8, assume water depth $D = 2m$, and simulate the I_c, D_c, and B_c for 5 color charts at distances $z = 1, 2, 3, 4, 5, 10$ meters.

What to include in your report:

1. Include, side-by-side, the 5 simulated color charts for water type 2, and for water type 8. Label them clearly. As before, you might want to white balance them relative to an unattenuated image at the surface.

2. For both water type 2 and 8, plot the values of I_c, D_c, and B_c versus z. Show each color channel with a different color.

Discuss:

1. What are the differences between the way the direct signal (D_c) attenuates in each water type?

2. What are the differences between the way the backscattered signal (B_c) grows in each water type?
3. What is the effect of the water depth D? What would happen if you simulated $D = 15m$ in water type 8?

3 Part B - Real World Data

3.1 Exercise 1: Extract RGB values from all charts

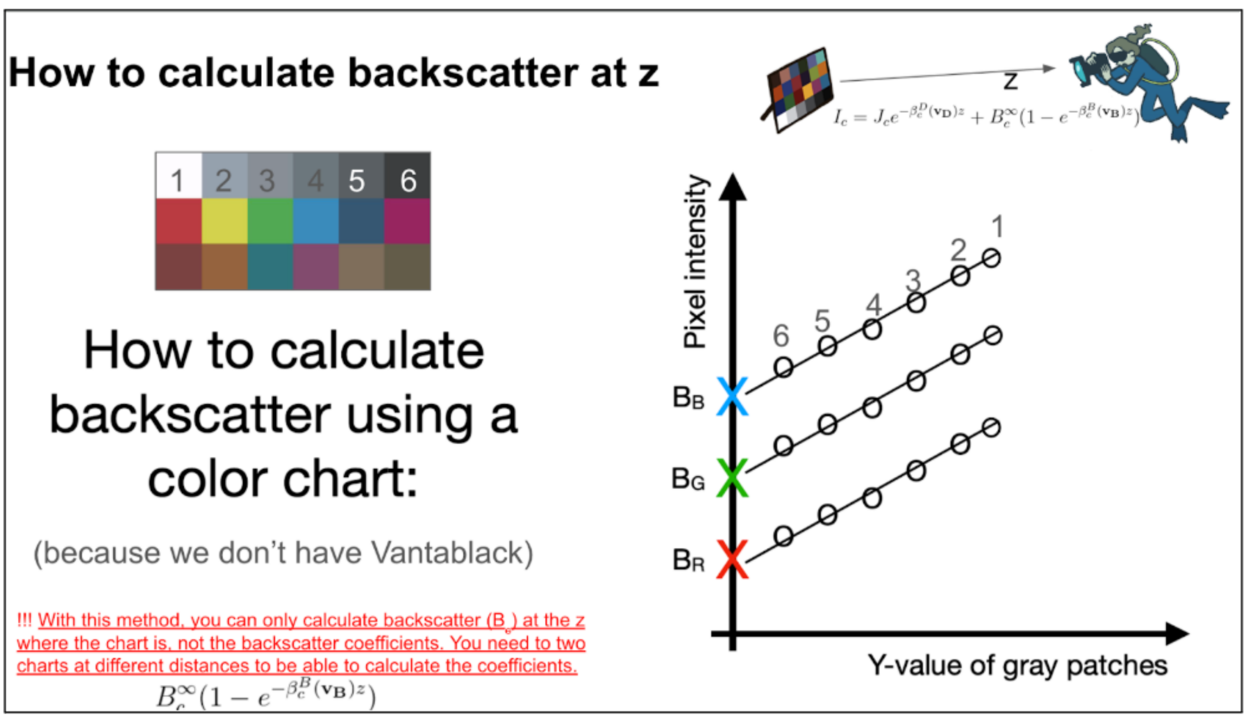
You have two linear images. These images are in the camera-specific RGB color space. From these images, extract RGB values for each patch of each color chart. Our underwater color charts have 18 patches. So let's say if you have 5 color charts in an image, you will extract 5 matrices of 18×3. Save these because you will need to reuse them.

3.2 Exercise 2: Extract D values from all charts

Using the D-images, extract the depth (range) value for each color chart. Here, there is no need to do it at patch level, because we can assume all patches on a given chart are at the same distance from the camera. So if you have 5 color patches in your scene, you will end up with a vector of 1×5. Save these because you will need to reuse them.

3.3 Exercise 3: Calculate backscatter at each chart

Following the method described in the lecture notes, calculate Backscatter B_c at each color chart. If you have 5 color charts in the scene, you should end up with 5 backscatter values, so a matrix of 5×3. Save these values.



What to include in your report:

1. For each image, a plot of B_c (y-axis) as a function of z (x-axis). Color code color channels (e.g., use red markers for the red channel, green markers for the green channel,

and blue markers for the blue channel). Label the plots clearly, so it's clear which goes with which image.

2. For each image, a visualization of the B_c values in the format of the color chart (should be a solid color).

3.4 Exercise 4: Subtract the backscatter to obtain the direct signal

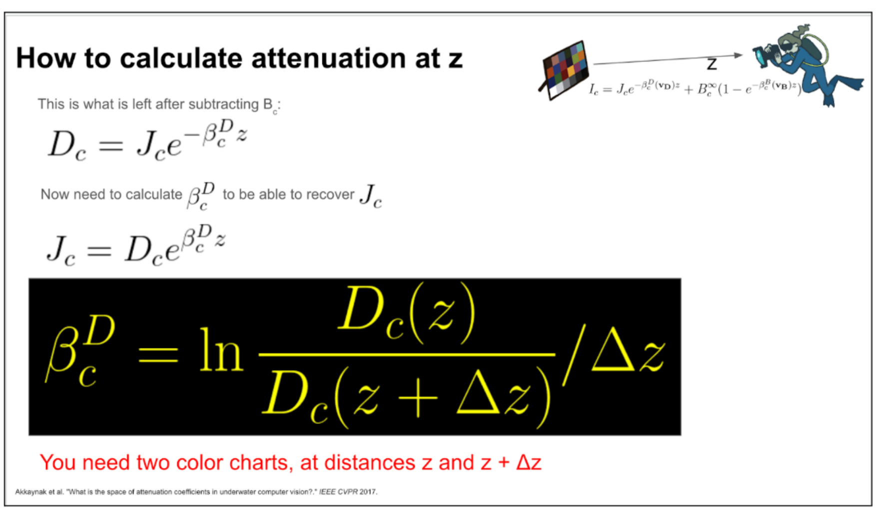
From the linear-RGB values you obtained in **Exercise 1** for each chart, subtract the B_c to obtain D_c.
What to include in your report:

1. For each image, a plot of D_c (y-axis) as a function of z (x-axis). Color code color channels (e.g., use red markers for the red channel, green markers for the green channel, and blue markers for the blue channel).

2. For each image, a visualization of the D_c values in the format of the color chart.

3.5 Exercise 5: Calculate the attenuation coefficient β_c^D for each color chart

For each image, following the method described in the lecture notes, calculate the attenuation coefficient β_c^D using pairs of color charts (one at distance z and one at distance $z + \Delta z$). Remember that β_c^D is a function of distance and it will have a different value depending on which pairs of charts you use.
Calculate it for several pairs of charts because due to the real-world nature of the data, it might work better with some pairs than others.
Each β_c^D will be a vector of size 1×3.



3.6 Exercise 6: Recover J_c, the image with attenuated colors, for each color chart

Using the attenuation coefficient(s) you got in **Exercise 5**, for both images, recover J_c as described in the lecture notes. For each chart, J_c will be of size 18×3. Note that these `RGB` values are still in your camera's unique color space.

What to include in your report:

1. A plot of J_c (y-axis) as a function of z (x-axis) (should be more or less constant across z). Color code color channels (e.g., use red markers for the red channel, green markers for the green channel, and blue markers for the blue channel).

2. A visualization of the J_c values in the format of the color chart.

3.7 Exercise 7: White balance the "J_c" images for each chart

White balance your resulting J_c image using any gray patch in any chart. This should be possible because at this point, you have compensated for all the effects of water due to z, and now you only need to correct global illumination.

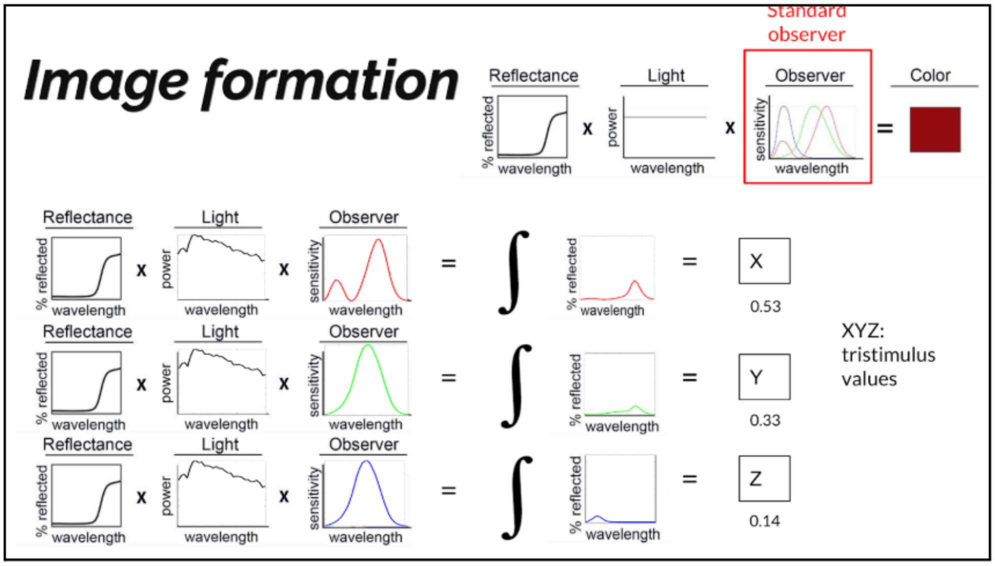
What to include in your report:

1. A plot of white-balanced J_c (y-axis) as a function of z (x-axis). Color code color channels (e.g., use red markers for the red channel, green markers for the green channel, and blue markers for the blue channel).

2. A visualization of the white-balanced J_c values in the format of the color chart. Remember to note with which patch you white balanced!

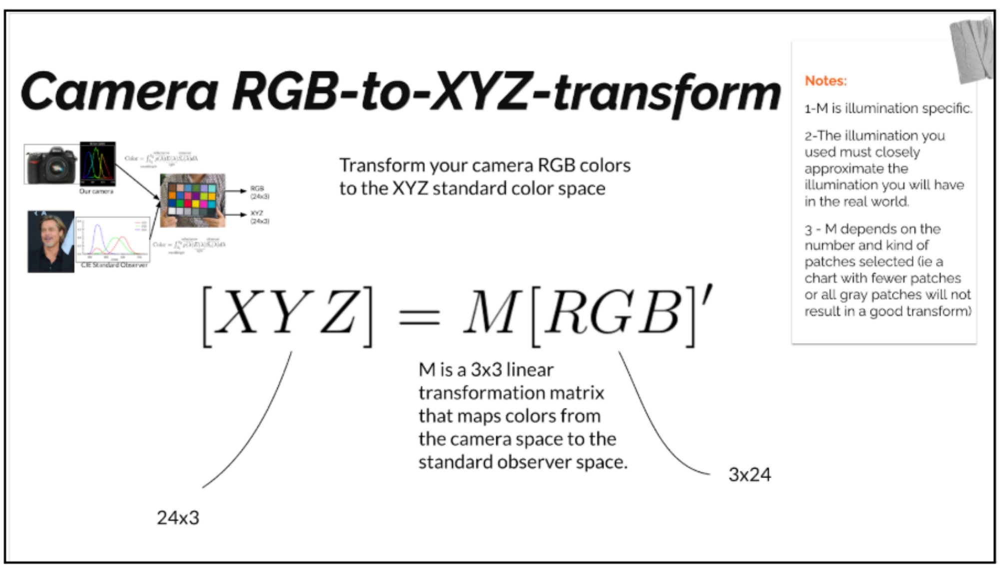
3.8 Exercise 8: Obtain XYZ values for the reference color chart that is used in the photos

Following the lecture notes, using $D65$ illuminant and the standard observer curves, obtain the XYZ values for this chart. Remember to white balance these values with the same patch you used in **Exercise 7**.



3.9 Exercise 9: Build a camera to XYZ color transform

In the course github, we have provided you the spectral sensitivities of the cameras that took both datasets (Eilat dataset with the Sony camera, and the Florida dataset with the GoPro). Simulate the RGB values these cameras would capture for the DGK color chart under the D65 illuminant. Remember to white balance these values with the exact same patch you did in the previous exercise. Build an $RGB2XYZ$ color transformation matrix M for each camera.



3.10 Exercise 10: Apply matrix M to bring your colors to the XYZ color space

Apply the appropriate M matrix to the J_c values you had obtained for each chart, in each image, and obtain the XYZ versions of your color chart RGB values.

3.11 Exercise 11: Convert colors from all charts in XYZ color space to sRGB

For both images, convert your chart XYZ values to sRGB color space. Recall that these conversion matrices are standard and can be found online. If all went well, at this point all your reconstructed J_c values should be identical.

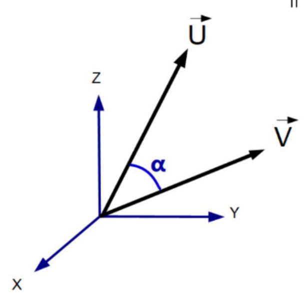
What to include in your report:

1. A plot of J_c in sRGB color space (y-axis) as a function of z (x-axis). Color code color channels (e.g., use red markers for the red channel, green markers for the green channel, and blue markers for the blue channel) OR use subplots per color channel. Try to overlay data from both images on top of one another. At this point, they should be identical.
2. A visualization of the sRGB J_c values in the format of the color chart. Remember to do proper white balancing so all charts can be visually compared.

3.12 Exercise 12: Calculate errors between different color charts in a given image

Compare, using the color angle error, the errors between **each patch across different charts**. For example: compute the error between white patch from chart 1 and white patch from chart 2, and then chart 3, etc (do all combinations). Compute the average error across all charts for the white patch. Repeat for all patches.

$$D(\mathbf{U},\mathbf{V}) = \angle(\mathbf{U},\mathbf{V}) \triangleq \arccos\left(\frac{\mathbf{U}\cdot\mathbf{V}}{\|\mathbf{U}\|_2\|\mathbf{V}\|_2}\right)$$



***Include in your report (Exercise 12)*:**
A plot of average errors (y-axis) per patch (x-axis). Repeat for both images. Feel free to include any other details or questions/observations in your report, or if you have suggestions for computing things differently. Underwater Colorimetry is a new and emerging field and there is a lot of work to be done!

THE END

If your name begins with 'A', try some of this analysis with grayscale images.

Extra credit of 10% and an offer to join the COLOR Lab: present results from all images from all datasets.