Improving Sea-Thru With Monocular Depth Estimation Methods

John Gibson
johngibson@wustl.edu

Abstract

A recent advance in underwater imaging is the Sea-Thru method, which uses a physical model of light attenuation to reconstruct the colors in an underwater scene. This method utilizes a known range map to estimate backscatter and wideband attenuation coefficients. This range map is generated using structure-from-motion (SFM), which requires multiple underwater images from various perspectives and long processing time. In addition, SFM gives very accurate results, which are generally not required for this method. In this work, we implement and extend Sea-Thru to take advantage of convolutional monocular depth estimation methods, specifically the Monodepth2 network. We obtain satisfactory results with the lower-quality depth estimates with some color inconsistencies using only one image.

1 Introduction

Underwater image datasets must be color-reconstructed for species identification, salvage operations, and other tasks requiring accurate visual information. Most datasets are manually color balanced using a color chart, requiring human interaction and the presence of a color chart in the scene. Recently, a method called Sea-Thru utilizing a physical model of light attenuation and backscatter in water was published [1], and gained both media and scientific attention. Sea-Thru reconstructs colors in a physics-accurate way and produces high-quality images that can be used as inputs to other machine learning algorithms or more easily analyzed by humans. However, this method uses depth maps generated using Structure-from-Motion (SFM), a technique that requires many images of the scene from different perspectives. This limits use of this technique to certain classes of datasets that either contain these images or already include depth information.

In order to address this shortcoming, we turn to recent advances in monocular depth estimation. These convolutional neural network-based methods use visual cues in the scene to produce a rough estimate of depth [2]. Although these methods are generally trained using above-water images, with appropriate preprocessing we can obtain a reasonable depth estimate for underwater images. This technique opens up Sea-Thru to applications where color correctness is not necessarily crucial, but a more accurate rendition of colors is desired.

Figure 1: Example of original image and recovered, color-corrected image from monocular depth estimation. Note the red color shading in some areas of differing illumination and the image artifacts near the border. However, the coloring is much improved.

In order to evaluate our method, we use the underwater RGBD dataset provided by [1] and estimate depth for these test images using monodepth2 [3], a monocular depth estimation framework by Godard *et al.* We then visually compare results from the two methods to determine color reconstruction accuracy, as a direct comparison without accurate depth information is quite difficult.

2 Background & Related Work

2.1 Underwater Imaging

Underwater imaging is inherently different from atmospheric imaging due to the nature of the medium. Although backscatter and wideband attenuation occur in media such as air, the extent to which these phenomena affect the image at normal depths is negligible. These phenomena severely affect the image underwater, as if taking an image through a thick, colored fog with uneven illumination – clearly a challenging setting. Previous models based on atmospheric imaging neglect the wavelength-dependency of the underwater setting and assume these coefficients to be the same, in contrast to the Sea-Thru model, which evaluates the effects independently.

In the Sea-Thru model, underwater image formation can be modeled as the following:

$$I_c = D_c + B_c \tag{1}$$

where c is an RBG channel, I_c is the observed image for the channel, D_c is the direct signal from the scene, and B_c is the effect of backscattering. As the medium has different attenuation coefficients and backscatter for each wavelength of light, we must calculate these values independently for each color channel. Values closer to red are more subject to attenuation, thus decreasing the intensity of the red channel and lending water its characteristic blue-green cast, and backscatter contributes heavily the cloudiness in underwater scenes. If we can describe the effect of backscatter and attenuation at each depth, we can reconstruct the image as follows:

$$J_c = D_c \exp(\beta_c^D(z)z) \tag{2}$$

where J_c is the true image, $D_c = I_c - B_c$, and $\beta_c^D(z)$ is the depth-dependent wideband attenuation coefficient. Backscattering is dependent on color channel c as well as depth and can be modelled as

$$\hat{B}_c = B_c^\infty \left(1 - \exp(-\beta_c^B z)\right) + J_c' \exp(-\beta_c^{D'} z) \tag{3}$$

where $\hat{B}_c$ corresponds to the contribution of the backscattering in the scene, B_c^∞ corresponds to the global veiling light of the scene, and $J_c' \exp(-\beta_c^{D'} z)$ corresponds to a residual term that behaves like the image (e.g. to account for patches that are not truly black).

Thus, the image formation can be expanded as:

$$I_c = J_c e^{\beta_c^D(z)z} + B_c^\infty \left(1 - e^{-\beta_c^B(z)z}\right) \tag{4}$$

2.2 Monocular Depth Estimation

Classical depth estimation pipelines use pairs of aligned stereo images. A physical model of disparity can therefore be calculated from corresponding parts of the aligned imges. The fundamental challenge of monocular depth estimation is that this physical model cannot be used, and therefore depth must be estimated from visual cues in the image. Many approaches exist for this task, most trained end-to-end using depth maps from stereo images. These can give a reasonable approximation of depth for many tasks, but generally only generalize well to images like the training sets.

Monodepth

A recent advance in monocular depth estimation is the development of an end-to-end self-supervised network for depth estimation. We briefly describe the method; for more details see [3]. The network takes in a pair of images of the scene from different viewpoints and trains both a depth and pose network based on those images. Instead of the classical minimization of the error in the depth maps, the network instead estimates the pose between the two images, computes the depth map, and then calculates the reprojection of the first image onto the second using the estimated pose and depth map. The loss to be minimized is the reconstruction error of this estimation. This allows the network to be trained purely from stereo images, instead of relying on complicated and expensive depth map estimates.

3 Method

3.1 Preprocessing Depth Maps

Structure-from-Motion Maps

The provided structure-from-motion maps have depths provided in meters. However, depth information is provided for only a portion of the scene, and the other depth values must be imputed. In addition, some areas of the depth map have invalid (e.g. negative) values. We first find the minimum and maxiumum depth values in the map. Then, for some user-defined percentage p, (default 1%), we set the values in the lowest p percent of depths to the maximum value in that range. We observe that missing values in the depth maps correspond to far-off areas, and thus we set the missing values to the maximum value of the depth map. This assumption produces images with inaccurate colors in some areas of the scene with missing values, but still generally offers improvement over the raw image.

Monodepth Maps

Raw underwater images are first fed through an adaptive histogram equalization pipeline before being used as input to the monodepth2 network. In contrast to depth maps determined from SFM and provided in the dataset, depth

maps from monodepth2 and other monocular methods are generally relative depths instead of absolute depths; e.g. depth values are dimensionless and only given in relation to other objects in the scene as opposed to absolute depths in meters. We must therefore estimate absolute depths from the provided relative depths. Visibility in water declines rapidly with distance, and therefore we define a maximum visibility in meters (default 10) by which to scale the relative depths. In addition, due to the field of view of the specific lens, elements at small depths are generally not visible when imaging the scene. To account for this, we simply add a value in meters (default 1) to each depth in the scene. These maps can then be used in the Sea-Thru algorithm.

As code for Sea-Thru is not provided by the authors, we first implement the method, with some modifications.

3.2 Estimating Backscatter

Backscatter values are estimated as in the Sea-Thru paper. The authors of Sea-Thru observe that image values corresponding to black or completely shadowed regions of the scene are entirely determined by backscatter, as there is no light reflected from the scene itself in those areas.

To estimate backscatter, the authors divide the region into evenly-spaced depth intervals and take the bottom 1% of RGB triplets in that interval. These pixels are then split into individual RGB values, and values for B_c^∞, β_c^B, J_c', and $\beta_c^{D'}$ (equation 3) are then estimated for each color channel using the corresponding depth values, thus giving a model for backscatter at all depths for each color channel. We used scipy's `curve_fit` function to perform this optimization and found that the method works well in practice when taking the parameters with minimum reconstruction error after 10 random restarts. However, we found data at very small depths to be quite noisy, and therefore we set a minimum cutoff for the depths of colors used in the estimations (default 0.1st percentile of the depth values).

3.3 Estimating Attenuation Coeffcients

Course Estimate

Although we estimated a single scalar value for β_c^D in section 3.2, in the Sea-Thru model the wideband attenuation coefficients take into account regions of the scene not in shadow, thus reflecting light to the sensor through the water medium. Therefore, we must refine our estimate of β_c^D to account for varying depths. Empirical results and simulations support modeling $\beta_c^D(z)$ as a two term exponential:

$$\beta_c^D(z) = a\exp(bz) + c\exp(dz) \tag{5}$$

where a, b, c, d are scalars. Recall from equation 2 that $J_c = D_c \exp(\beta_c^D(z)z)$, and thus:

$$\begin{aligned} \frac{\log J_c - \log D_c}{z} &= \beta_c^D(z) \\ \frac{-\log E_c}{z} &= \beta_c^D(z) \end{aligned} \tag{6}$$

Where E_c corresponds to the illuminant map of the scene. This allows us to obtain a rough estimate of $\beta_c^D(z)$ for the scene.

Estimating Illuminant Map

The authors of Sea-Thru use local space average color (LSAC) to esstimate the illuminant map using a range map. This consists of updating the following equations for each pixel location (x, y):

$$a'(x,y) = \frac{1}{|N_\epsilon(x,y)|} \sum_{(x',y')\in N_e(x,y)} a_c(x',y') \tag{7}$$

$$a_c(x,y) = D_c(x,y)\, p + a_c'(x,y)(1-p) \tag{8}$$

where p controls the locality of the neighborhood (default 0.01), and $N_\epsilon(x,y)$ is the neighborhood of (x,y) such that

$$N_e(x,y) = \{(x',y') \;:\; |z(x,y) - z(x',y')| \le \epsilon\} \tag{9}$$

Then $\hat{E}_c = f a_c$, where f is a constant depending on scene geometry and controls global illumination. The authors of Sea-Thru suggest $f = 2$. In addition, we set the minimum size of a neighborhood to be 50 pixels, and reassign smaller neighborhoods to the closest neighborhood of large enough size. In addition, we perform morphological closing on the neighborhood map with a square structural element of side length 3 to remove holes and refine edges. We also perform bilateral filtering on the illumination map to smooth regions inside neighborhoods while paying attention to neighborhood boundaries.

Refined Estimate

To refine the estimate, the authors of Sea-Thru suggest minimizing the reconstruction error of depths as follows: first, rewrite equation 6 as

$$\hat{z} = \frac{-\log E_c}{\beta_c^D(z)} \tag{10}$$

and finding the values of a, b, c, d that satisfy

$$\min_{\beta_c^D(z)} \|z - \hat{z}\| \tag{11}$$

We formulate the problem as suggested, but failed to achieve consistent convergence. Instead, we minimize

$$\min_{a,b,c,d} \|\beta_c^D(z) - \hat{\beta_c^D}\| \tag{12}$$

with 10 random restarts, bounding a, b, c, d to obtain decaying exponentials, and taking the parameter set that minimizes the reconstruction error in equation 11. This has good convergence performance and performs well in practice. If this still does not converge, we use a linear model instead of the two-term exponential. We also add a multiplier l to counteract the overestimation from the illumination map's value of f (default 0.5). In addition, we filter the input data such that successive data points are at least 1% of the full depth range away from each other; this balances the data set such that an exponential trend can be estimated instead of being dominated by the clusters of datapoints at minimum and maximum depth.

3.4 White Balancing

After reconstructing the image using equation 2, we perform white balancing. Sae-Thru uses the Gray World Hypothesis assumption for white balancing, but we find a more visually appealing image from calculating the channel scaling factors as the inverse of the average of the top 10% of the values in each channel.

4 Experimental Results

4.1 Comparison of Results

We compare the estimated values for parameters on the same image in figures 2 and 3, and compare the final images in figure 4. The results are visually similar for both depth estimation methods. In addition, example neighborhood maps, lighting maps, etc. are shown in figure 5.

4.2 Success and Failure Cases of Monocular Depth Estimation

In images with depth maps that smoothly vary or vary by small amounts, monocular depth estimation (coupled with semi-accurate estimation of min and max depths by eye) can give results that are very similar to the true values. An example of this is shown in figure 6. This is mainly due to the smoothness of the estimated depth map and the estimation of minimum depth instead of and intrinsic strength of the network. In other cases where the image is of a scene where multiple objects and the sea floor are visible, the network also gives reasonable results. This is likely due to the similarity of the scene to stereo training data.

In scenes with little depth information or those very different from a scene above water, however, the network produces considerably less appealing images. Although the adaptive contrast stretching helps, the scene is simply too attenuated and unfamiliar to produce a good estimate.

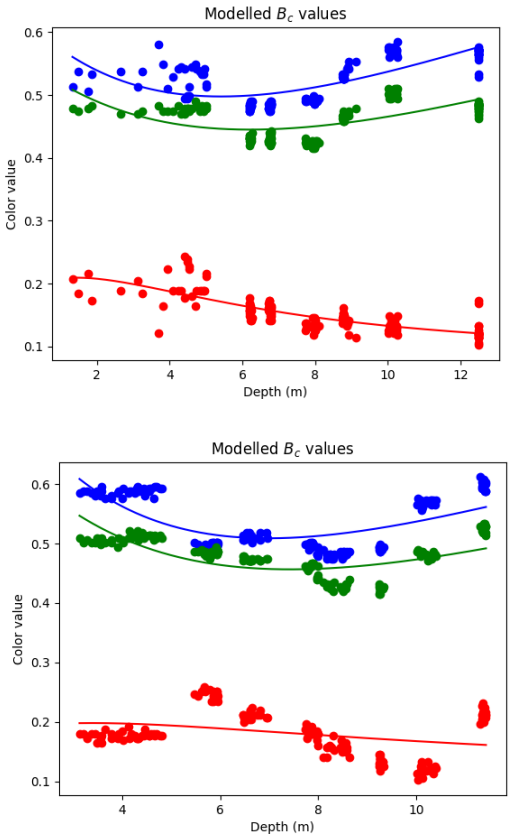

Figure 2: Comparison of estimated $B_c(z)$ values between SFM depth maps (top) and monodepth maps (bottom). The results are quite similar, lending support to this method.

5 Conclusion

In this work, we implement and extend the Sea-Thru method to take advantage of monocular depth estimation. Although this does not produce as accurate results as using depth maps from Structure-from-Motion, monocular depth estimation methods can produce visually appealing results on real data with minimal user interaction. We show samples of images computed using both methods and discuss where monocular depth estimation fails to produce accurate results.

In the future, we hope to enable the training of monocular depth estimation networks on this kind of color-corrected underwater scene data to hopefully improve the accuracy of this method in underwater and heavily obscured scenes. In fact, iterative rounds of color correction and retraining of the monocular depth network on this data could rapidly improve performance on underwater scenes.

We hope that this method proves useful to those wishing to view the beautiful colors of coral reefs and ocean floors without the use of specialized equipment.

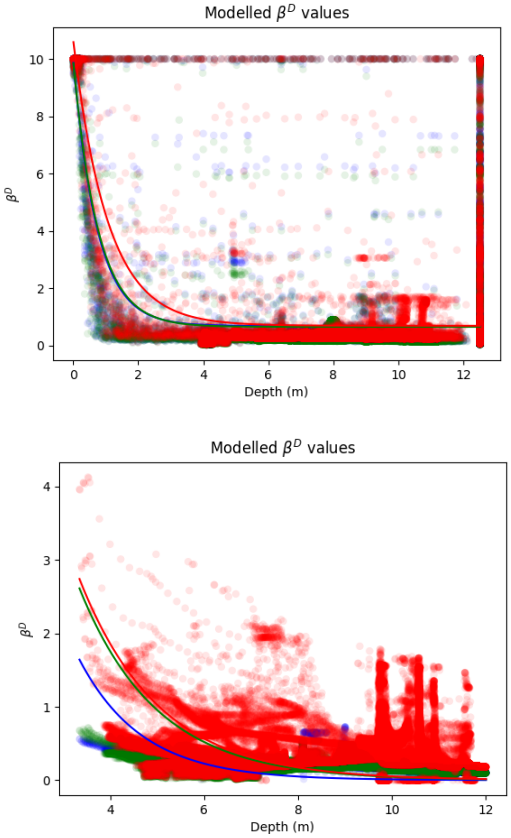

Figure 3: Comparison of estimated $\beta_c^D(z)$ values between SFM depth maps (top) and monodepth maps (bottom). Although the monodepth values are generally noisier, an exponential curve is still produced, and the final results are still visually similar.

Figure 4: Comparison of final images using SFM (top) and monodepth2 (bottom). Images were run with identical parameters, and min and max depths for monodepth were estimated using the SFM depth map to increase correspondence between the images. Both images have had exposure increased by 1 stop for the purposes of comparison. Artifacts due to inaccurate depth are present in each image; in the SFM image, depth information is only present for the seafloor and part of the reef. One can see where this information ends by the thin black line on the top image. Areas with depth values from SFM have excellent reconstruction, but areas without depth information were set the the maximum depth in the image, and have color artifacts. The monodepth map also has color artifacts from inaccurate depth values.

Acknowledgments

Thank you to the original authors for providing the training data and Gabriel Brostow and Niantic, Inc. for releasing and maintaining monodepth2.

References

[1] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 04 2019.

[2] Amlaan Bhoi. Monocular depth estimation: A survey, 2019.

[3] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

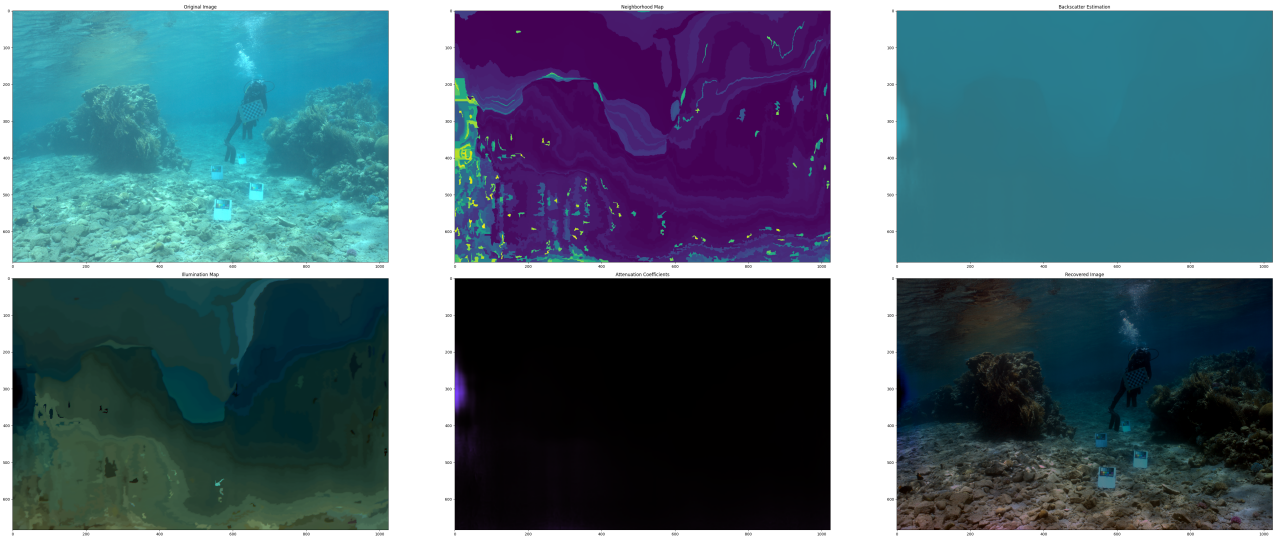

Figure 5: From left-to-right, top-to-bottom: input, neighborhood map, backscatter coefficients, illumination map, attenuation coefficients, and unprocessed output for a monodepth-processed image.

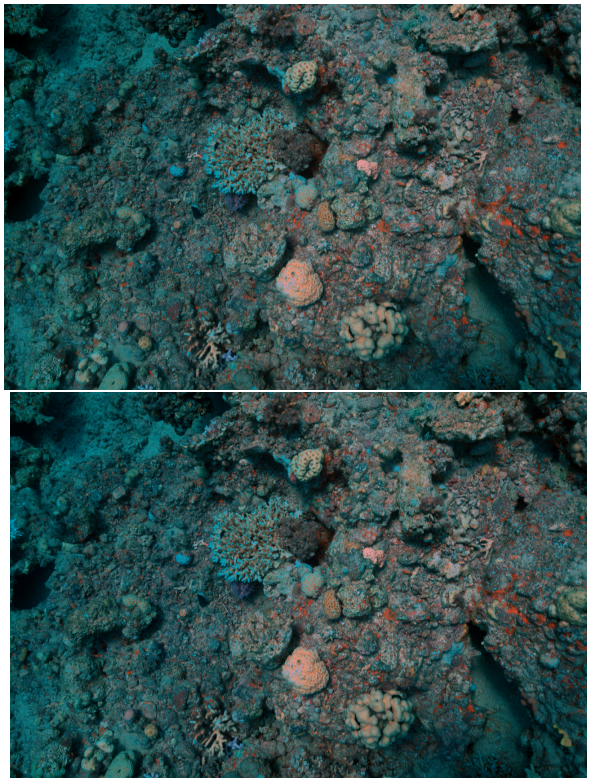

Figure 6: An example where both methods produce the same results given accurate min and max depth values. SFM is top, monodepth is bottom.