DeepCaustics: Classification and Removal of Caustics from Underwater Imagery

Timothy Forbes, *Student Member, IEEE,* Mark Goldsmith, *Member, IEEE,* Sudhir Mudur, *Sr. Member, IEEE,* and Charalambos Poullis, *Member, IEEE*

Abstract

Caustics are complex physical phenomena resulting from the projection of light rays being reflected or refracted by a curved surface. In this work, we address the problem of classifying and removing caustics from images and propose a novel solution based on two Convolutional Neural Networks (CNNs): SalienceNet and DeepCaustics. Caustics result in changes in illumination which are continuous in nature, therefore the first network is trained to produce a classification of caustics which is represented as a saliency map of the likelihood of caustics occurring at a pixel. In applications where caustic removal is essential, the second network is trained to generate a caustic-free image. It is extremely hard to generate real ground truth for caustics. We demonstrate how synthetic caustic data can be used for training in such cases, and then transfer the learning to real data. To the best of our knowledge, out of the handful of techniques which have been proposed this is the first time that the complex problem of caustic removal has been reformulated and addressed as a classification and learning problem. This work is motivated by the real-world challenges in underwater archaeology.

1 INTRODUCTION

In recent years, advances in computer vision have greatly impacted underwater archaeology [1], and exploration in general. Computer vision techniques are increasingly being used for recording and documenting underwater sites, and most often involve the capture and subsequent automated processing of images in order to generate an accurate 3D reconstruction of the site and/or panoramas in cases where the site is spread over a large area.

The automated processing traditionally involves extracting and describing distinctive features in each of the images, matching them, and then applying a combination of Structure-from-Motion(SfM) and Multi-View Stereo(MVS) techniques. In the context of underwater archaeology, this has been shown to produce very good results [2] especially in deep waters i.e. $> 40m$, where no natural light reaches the site. In shallow waters i.e. $< 10m$, the same technique almost completely fails because of natural light coming in from above the water surface causing caustic effects on the site. Fast changes in illumination caused by the rapid motion of the water surface adversely affect feature detection and matching. As a result, all subsequent processing produce erroneous results.

In this work, we address the complex problem of removing the effects of caustics by first classifying and then removing them from the images. We propose a novel solution based on two

T. Forbes, M. Goldsmith, S. Mudur, C. Poullis are with the Department of Computer Science and Software Engineering, Concordia University, Quebec H3G 1M8 , Canada

Manuscript received June 1, 2017;

small and *easily trainable* CNNs. Real ground truth for caustics is not available and is also hard to generate. We show how a small set of synthetic data can be used to train the network and later transfer the learning to real data with robustness to intra-class variation. The proposed solution results in caustic-free images which can be further used for other tasks as may be needed. To the best of our knowledge, this is the first time that the complex problem of caustics removal is reformulated and addressed as a classification problem. All prior techniques rely on low-level image enhancement and do not address the difficult problem of pixel level classification.

Our technical contributions are:

- SalienceNet: A small and easily trainable CNN architecture for the classification of caustics. The network is trained, with input consisting of synthetic images containing caustics and the corresponding masks as ground truth, to produce saliency maps of the likelihood of caustics occurring at each pixel.
- DeepCaustics: A CNN architecture for the removal of caustics. The network is trained with input pairs of the synthetic images containing the caustics and the corresponding saliency maps [generated by SalienceNet], to produce a caustic-free image.
- A method for training the network on synthetic caustic data and transferring the learning to real data.

The paper is organized as follows: Section 2 briefly reviews the state-of-the-art in the area. Section 3 provides an overall description of the way this system got developed. In Section 4 we present the design process and the final network architecture. Section 5 we discuss the results and network performance and we conclude in Section 6.

2 Related Work

Although a plethora of work has already been reported on caustics generation, only a handful of techniques have been proposed for caustics removal; the majority within the context of image enhancement. Below we review some of the work that best relates to the caustics removal problem.

In [3] the authors present a technique for tuning a sunlight-deflickering filter for moving scenes underwater. They propose a continuous parameter optimization inside a basic filter, which employs feedback in order to improve the performance. As reported in their paper there is a high sensitivity of the filter's performance to badly optimized parameters and in particular, the segmentation parameter which is part of the objective function in the optimization.

A different approach was presented in [4]. The authors present a mathematical solution which involves calculating the temporal median between images within a sequence. A strong assumption of this work, is the fact that feature matching [Harris corner detection variant in [5]] is employed for the formation of the sequence which makes this approach very susceptible to the light variations in the images and in particular caustics effects. The authors later extend their work in [6] and propose an online sunflicker removal method which treats caustics as a dynamic texture. As reported in the paper this only works if the seabed or bottom surface is flat. Similar approaches have also been proposed for general cases of dehazing and descattering of images such as [7], [8], [9].

The authors in [10] propose a method based on processing a number of consecutive frames. These frames are analyzed by a non-linear algorithm which preserves consistent image components while filtering out fluctuations. Their proposed method however does not take into account the camera motion which almost always leads to registration inaccuracies.

In order to avoid registration inaccuracies the authors in [11] present a method for removing caustics using a stereo-rig. The stereo cameras provide depth maps which can then be registered

together using ICP (Iterative Closest Point; a technique for aligning multiple geometries to one another). This again makes a strong assumption on the rigidity of the scene which is seldom the case in underwater.

In [12], the author propose an approach which employs optical flow techniques and curvature predictions of pixel traces during the motion. As with the aforementioned technique there are strong assumptions on the small motion and color constancy between consecutive frames.

Structure from motion techniques such as [13] and [14] have also been explored for reconstructing underwater scenes with poor results. Although the scene is static and the camera is moving smoothly the variation in illumination throws off the feature matching algorithm.

It is clear from the above that trying to procedurally solve the problem of caustics classification in images would require simplifying assumptions on the many varying parameters involved, such as scene rigidity, camera motion etc. In this paper we propose a method which makes no such assumptions. Each image in a sequence is processed individually and as it is reported in Section 5 the results are consistent between consecutive images although no temporal information is taken into account.

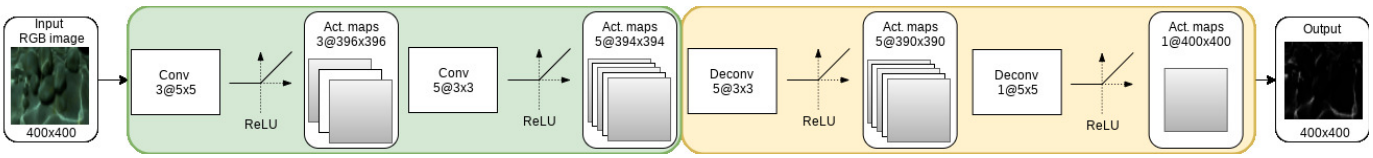


Fig. 1: SalienceNet: a 4-layer CNN consisting of 2 convolutional layers followed by 2 deconvolutional layers. A ReLU activation unit follows each [de-]convolution operation. All [de-]convolution kernels have size 3×3.

3 Network Training Experiments

Caustic detection and removal from real images was attempted using a Convolutional Neural Network (CNN) with an architecture similar to that of a Convolutional Autoencoder reported in [15], where the first layers perform convolution, and subsequent layers perform de-convolution. CNNs are a natural choice when dealing with images; however, unlike traditional convolutional networks, which rarely attempt pixel level classification, we require an architecture that produces images of sizes matching input image sizes, hence the need for de-convolution layers.

Since there is no ground truth data available for real world underwater caustics, we decided on supervised learning using synthetic data for ground truth. Using sample underwater video images with caustics we created a set of synthetic data using 3D objects for underwater seabeds, multiple lighting and global illumination for rendering caustic effects with the virtual camera located below the water surface as in real world shallow water imaging. In addition to the rendered caustics images, we rendered the masks containing confidence values of a caustic occurring at a pixel and also corresponding caustic-free images. Using this synthetic data set (80% training, 20% testing), we were able to obtain very good results for confidence learning using the small neural network described in the next section.

It was interesting to see that the network seemed somehow to be picking up the complex structure and illumination/colour pattern of caustics and surrounding regions. We also noted that the colour variation in our synthetic data was different from that of the real world videos. So, for transferring the learning to real world, we decided to apply a colour transfer operation to real world videos getting them into similar illumination and color space. We then added a second component, DeepCaustics, for learning to remove caustics. The final network architectures, experiment parameters, results etc. are described in the following sections.

4 NETWORK ARCHITECTURE

As mentioned earlier, our proposed solution consists of two CNNs, SalienceNet and DeepCaustics as presented in the following sections.

4.1 SalienceNet

The input to SalienceNet is a rendered RGB image containing caustics of an underwater scene. The network operates on a batch of 32 images of size 400×400. Each pixel of the output image takes a value in the range of $[0, 1]$, corresponding to the confidence of caustics occurring at that pixel. After extensive experimentation, we have concluded that the network architecture with the optimal performance consists of four hidden layers; the first two consisting of 3 and 5 convolution filters respectively, and the last two consisting of 5 and 1 de-convolution filters respectively. The filter sizes are 5×5, 3×3, 3×3, 5×5 in each layer respectively. This results in a total of $2 \times (5 \times 3 \times 3) + (4 \times 5 \times 5)$ weight parameters and $8 + 6$ bias parameters, for a total of 204 parameters to be learned. After each convolution/deconvolution in the network follows a ReLU activation unit [16]. Initially, sigmoid activation units were used in the last layer to ensure that the final output is in the range $[0, 1]$ however, our experiments have shown that ReLU units perform better [they still map the output in the range $[0, 1]$ provided the input data falls within the manifold learned] and, in addition computing the gradients becomes more stable during back-propagation i.e. no 'squashing' leading to vanishing gradients. Adding more units and/or more layers has also been tested, but with no noticeable improvements. Larger filter sizes were also tested, but yielded blurry results. In order to get reasonable results with an initial layer consisting of larger filters, more layers with decreasing filter size were needed, but this required a reduction in the size of the images in the data set, due to memory constraints, and added no significant advantages. A diagram of the network's architecture, chosen based on all the experimental evaluations and considerations described above, is shown in Figure 1.

Thus, the network models the following operation,

$$ReLU(\Psi^{-1} * ReLU(\Psi^{-1} * ReLU(\Psi \star ReLU(\Psi \star X)))) \rightarrow \Phi^{p} \tag{1}$$

where p is a pixel and Φ is its saliency value. In the above equation X denotes the input data, Ψ denotes a convolution kernel, $\star$ denotes the convolution operation, Ψ^{-1} denotes a de-convolution kernel, $*$ denotes the de-convolution operation, and $ReLU(.)$ is the rectified linear unit activation function.

As previously mentioned, the output is a gray-scale saliency map with pixel values ranging from $[0, 1]$; the larger the saliency value the higher the confidence of caustics occurring at that pixel.

4.1.1 Training and Transfer Learning - Caustics Saliency

The network was implemented and trained using theano API [19] on a set of 500 synthetic images. 60 synthetic images were reserved for validation. Figure 3 shows example pairs of input and ground truth images used for training the SalienceNet. The left column shows the rendered RGB image used as input and the center column shows the rendered caustics masks using global illumination and final gathering. Indirect lighting results in almost all pixels having a non-zero brightness value. The right column shows the thresholded caustics mask used during training as the ground truth saliency map. The dataset was created with Arnold Renderer using photon mapping [20] in Autodesk Maya 2017.

Although a synthetic data set was used for training, we were ultimately interested in the network's performance on the real data set which can be considered a form of transfer learning. The network was trained for 1500 epochs at a learning rate of 0.001, which required approximately

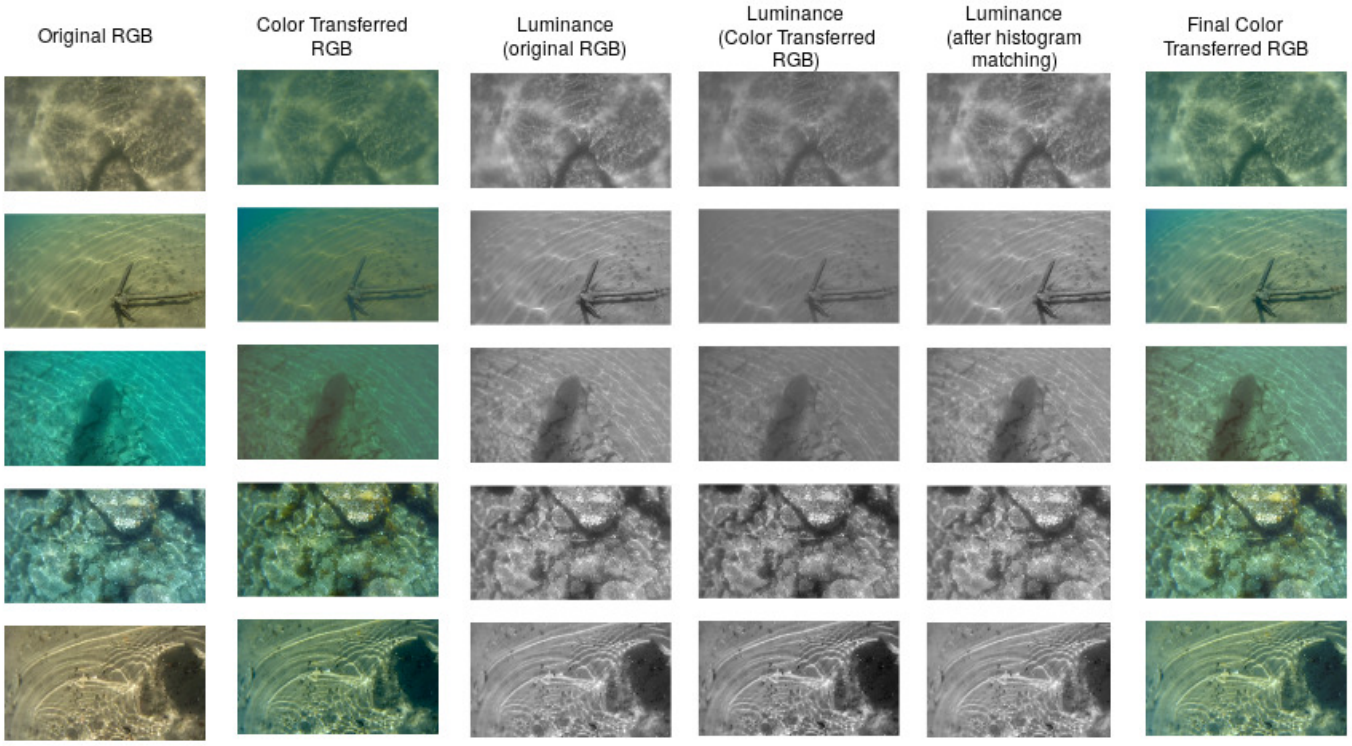


Fig. 2: Color transfer [17] and histogram matching [18] of the luminance channel are applied to the real images in order to make the real images fall within the manifold learned from the training images and remove possible bias to colors. The images showing the luminance are shown as grayscale images.

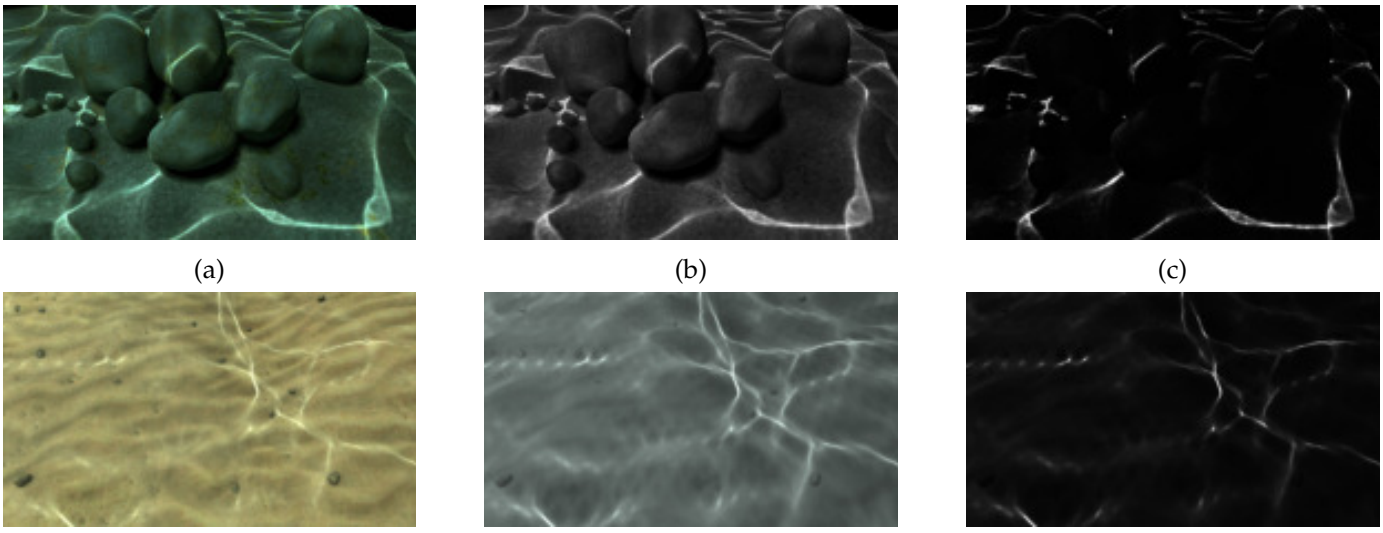


Fig. 3: Examples of the training data. Left column: The rendered RGB image used as input to SalienceNet and DeepCaustics. Center column: The rendered caustics masks using global illumination and final gathering. Indirect lighting results in almost all pixels having a non-zero brightness value. Right column: The thresholded caustics mask used during training as the ground truth saliency map.

9.5 hours of training on an GeForce GTX 1070 8GBVRAM GPU, followed by an additional 5000 epochs at a learning rate of 0.0001. A batch size of 32 was used. We also experimented with dropout [21], but no improvements were noticed, which indicates that overfitting was not a problem during the training process. Figure 4 shows the cost function error, modeled as a Mean-Squared Error (MSE) for 6500 epochs. Mean-Squared Error is the sum of squares of difference in pixel values from the ground truth and our predicted results. The formula is as follows,

$$\frac{1}{n}\sum_{j=1}^{n}(p_{ij}^{truth} - p_{ij}^{pred})^2 \tag{2}$$

where p_{ij} is a pixel in the i^{th} row and j^{th} column in the ground truth image or the SalienceNet image result.

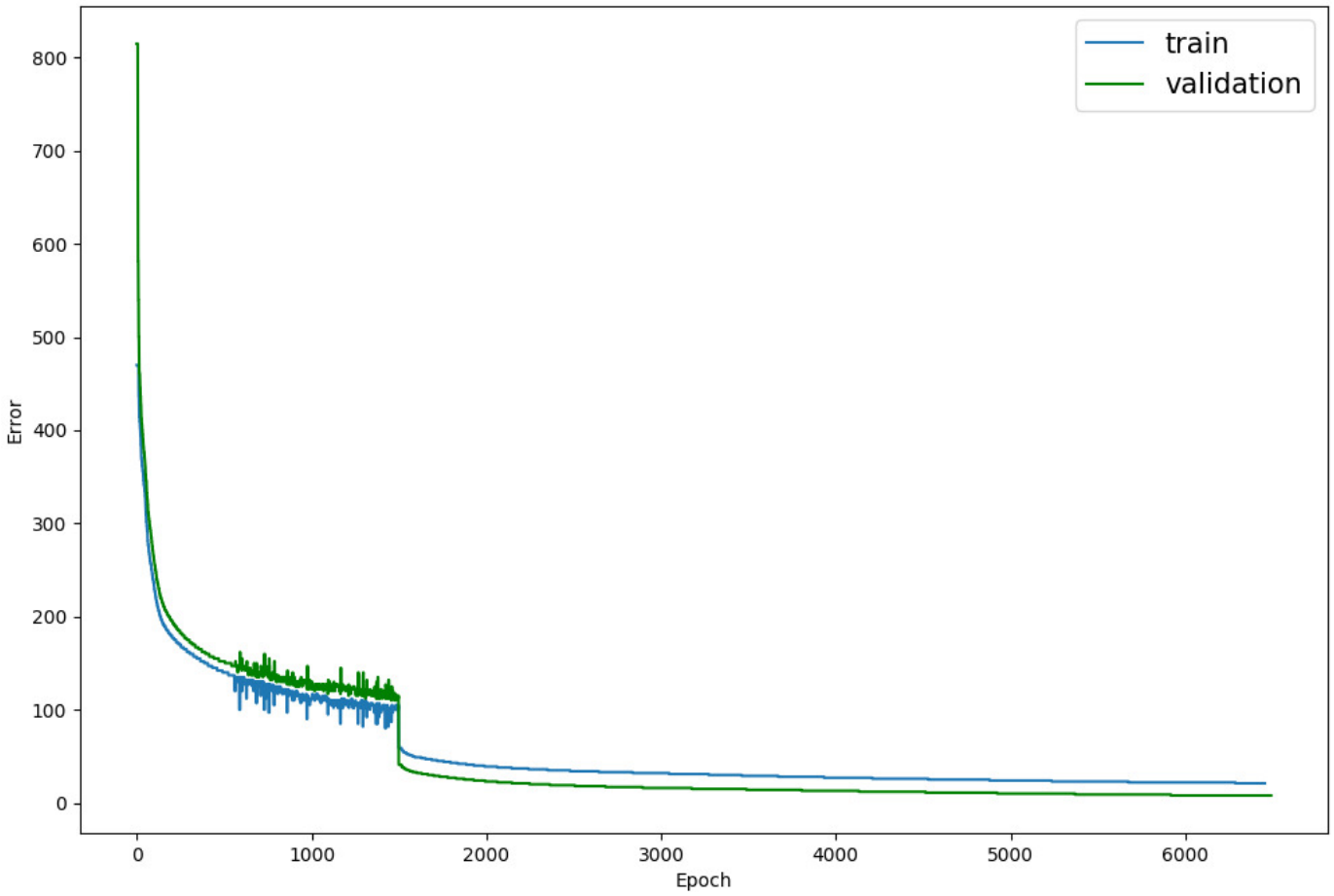


Fig. 4: SalienceNet Cost Function Error (training and validation curves are almost identical): First 1500 epochs at a learning rate of 0.001 and the next 5000 epochs at a learning rate of 0.0001. The cost function used is Mean-Squared Error (MSE).

As previously mentioned our main goal is to transfer the learning to real data sets. Towards this goal of transfer learning, due to the color variance of the water, seabed and caustics, the real images are first preprocessed such that the color space manifold matches that of the training set. Given a real image the preprocessing involves performing color transfer as in [17], converting from RGB color space to Luv, performing histogram matching as in [18] on the luminance channel L, and converting back to RGB color space. Figure 2 shows an example of this process.

Following the training of SalienceNet using synthetic data, we processed a number of varied real world underwater videos through this network and obtained very good results. Although

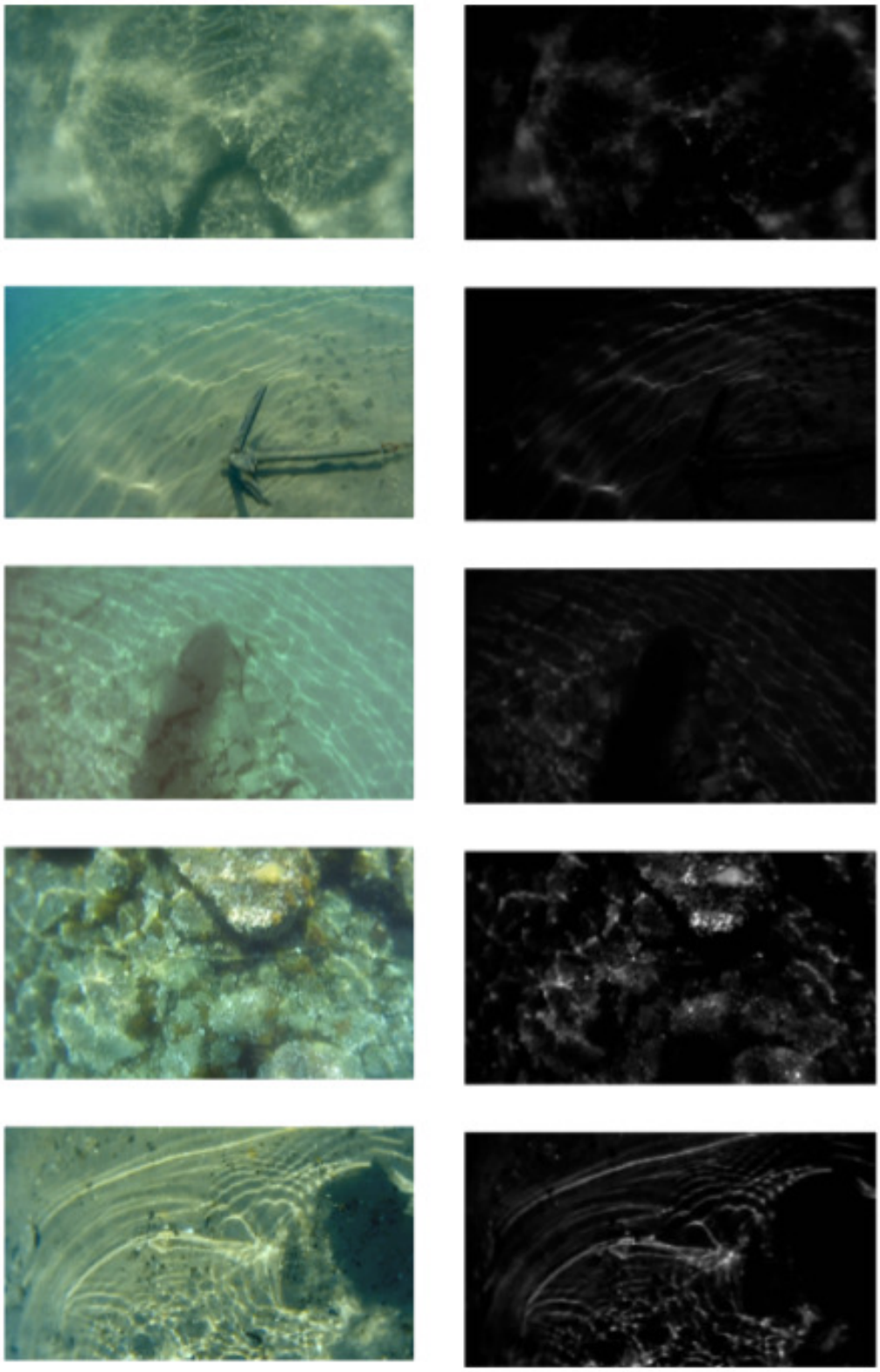


Fig. 5: (right) The saliency maps generated by SalienceNet for the five real images (left) from distinctive videos taken underwater containing caustics of varying frequencies and shape. Please refer to supplemental video for a higher resolution visualization.

we cannot directly assess the accuracy of the SalienceNet in removing caustics given the lack of ground truth on the real images, we do evaluate its performance in terms of the reconstruction

metrics described in Section 5.2. In other words, the metrics used to evaluate the reconstruction serve as a proxy for determining the performance of the networks. Five images from distinctive videos taken underwater are shown in the left column Figure 5 and the resulting saliency maps produced by SalienceNet are shown in the right column.

4.2 DeepCaustics

(a) (b) (c)

Fig. 6: Examples of the training data for DeepCaustics. (a) The rendered RGB image used as input to DeepCaustics. (b) The saliency map generated by SalienceNet. These two images [(a), (b)] become the input to the DeepCaustics network. (c) The ground truth used for training is a rendered caustic-free image corresponding to (a),(b).

The input to DeepCaustics is the pair of an image containing caustics and the saliency map generated by SalienceNet. The two are first coupled together into a 4-channel RGBA format where the fourth channel contains the saliency value for the corresponding pixel. The ground truth used for training is a rendered caustic-free image corresponding to the synthetic input images. An example is shown in Figure 6. The network operates on a batch of 16 images of size 400×400. The output of the network is a caustic-free RGB image corresponding to the input. After extensive experimentation, we have concluded that the network architecture with the optimal performance consists of six hidden layers; the first three consisting of 4, 2, and, 7 convolution filters respectively, the last three consisting of 7, 2, and 3 de-convolution filters respectively. The filter sizes are 3×3, 7×7, 3×3, 3×3, 7×7, 3×3 in each layer respectively. This results in a total of $(4 \times 3 \times 3) + 2 \times (2 \times 7 \times 7) + 2 \times (7 \times 3 \times 3) + (3 \times 3 \times 3)$ weight parameters and $(4 + 2 \times 2 + 2 \times 7 + 3)$ bias parameters, for a total of 410 parameters to be learned. Similarly to SalienceNet, after each [de-] convolution in the network follows a ReLU activation unit [16]. Figure 7 shows the architecture of the DeepCaustics network. An important observation first reported by [22] which was also confirmed during our experimentation is that scaling up the number of activation maps while scaling down the kernel size produces considerably better results. We have trained the DeepCaustics network on the same hardware as for SalienceNet.

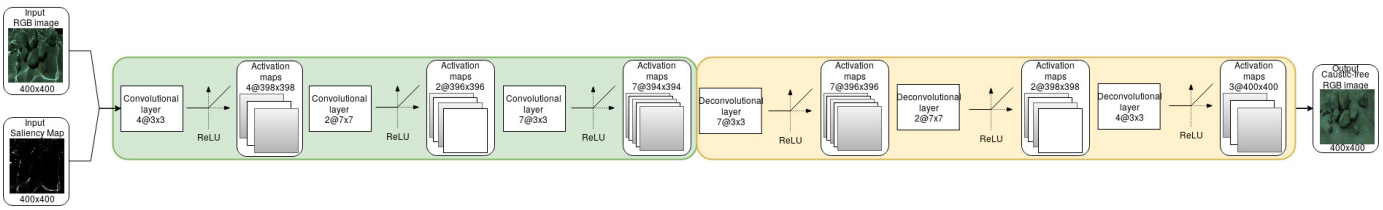


Fig. 7: DeepCaustics: a 6-layer CNN consisting of 3 convolutional layers followed by 3 deconvolutional layers. A ReLU activation unit follows each [de-]convolution operation.

4.2.1 Training and Transfer Learning - Caustics Removal

The colour transfered pre-processed images along with the saliency values generated by SalienceNet form the input for the trained DeepCaustics. Again, the expected output is a caustic-free image. The network is trained using 180 synthetic image-pairs, and 20 synthetic image-pairs for validation. Figure 8 shows the cost function error, modeled as a structural similarity index (SSIM); a quality measure of one of the images being compared, provided the other image is the ground truth. SSIM evaluates images accounting for the fact that the human visual system is sensitive to changes in local structure. SSIM is implemented by creating patches from the ground truth images (y) and the predicted images generated from DeepCaustics (x). For each patch we calculate the SSIM:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

such that μ is the average value of the patches, σ^2 is the variance, σ_{xy} is the covariance between patches x and y and σ is the standard deviation of the patches.

Our experiments have shown that it performs better than MSE in terms of network training as it has been also discussed in [23]. The network was trained for 2700 epochs with a learning rate of 0.001. The choice of 2700 epochs is somewhat arbitrary; the learning curve shows that much of the training progress occurs within the first 80 epochs.

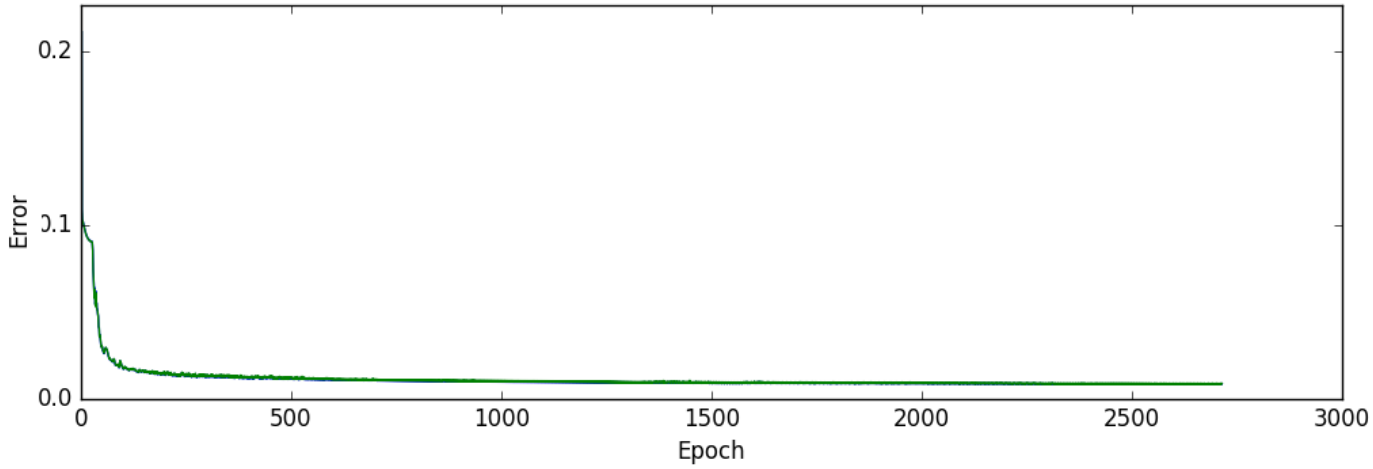


Fig. 8: DeepCaustics cost function error. The cost function used is Structural Similarity index (SSIM). SSIM is a quality measure of one of the images being compared, provided the other image is the ground truth. SSIM evaluates images accounting for the fact that the human visual system is sensitive to changes in local structure.

5 Discussion of Results

5.1 Network Parameter

In order to find the right parameters for our networks we went through several trials. We ran our networks for 200 epochs and selected the best parameters based on error, prediction results, and number of parameters. Our errors can be seen in table 1 with bold values representing our selections of each network.

When selecting our SalienceNet we noticed improvement when increaseing layers and layer complexity however the trade off between the error and the number of parameters was too large to ignore. We also noticed that our last two tests, while having low error, over fit towards the

Activation	Kernel	SalienceNet (MSE)	DeepCaustics (SSIM)
3,3	3,3	86.56	0.1628
3,5	5,3	**94.50**	0.1265
3,5,10	15,7,3	103.85	0.1947
4,2,7	3,7,3	115.56	**0.0361**
5,10,20	5,9,5	91.06	0.0359
5,10,20	15,9,5	116.55	0.0251
5,12,24	27,13,5	97.32	0.0245
20,10,15	3,7,5	71.06	0.0120
20,10,15	5,13,7	76.66	0.0179

TABLE 1: Errors for different network parameters after 200 epochs. Note that SalienceNet and DeepCaustics have different error functions, hence the different ranges of values. Activation column represents the feature maps for each layer with respect to the Kernel values squared for the kernel size. The first row would be a network of two layers, with the first layer consisting of 3 activation maps and a kernel size of 3x3. The second layer would also have 3 activation maps and a kernel size of 3x3.

artificial data producing erroneous salience images for real images. A similar issue happened with our DeepCaustics network. Our tests removed inefficient parameters but also permitted us to observe how differences in kernel size and how depth affected our results. Again increased complexity provided lower error but this meant more parameters and often over fitting.

The proposed approach has been extensively tested on real videos of underwater scenes containing caustics. The videos contain a large variability in terms of the color, the frequency of the caustics i.e. wave speed and formations, the background i.e. geometry and color [sand, rock, barnacles, etc], and shape i.e. wave interference.

As previously mentioned this work is motivated by the real world challenges in underwater archaeology. Processing videos for underwater entities in the presence of caustics is almost impossible. Figure 9 demonstrates that using the proposed approach successfully removes the caustics from the images therefore allowing the further processing. In the first column we show five sample frames from real world RGB videos containing caustics of varying characteristics. When these images are input to DeepCaustics along with their saliency maps generated by SalienceNet the results obtained are shown in Figure 5. The second column shows the five RGB images after color transfer and histogram matching. In the third column we show the results generated by DeepCaustics on the original images i.e. *without* any color transfer and histogram matching operations. As expected the network does not perform well for any images which fall outside the learned manifold. The last column shows the caustic-free images generated by DeepCaustics on the color matched RGB images shown in the second column. The caustics have been successfully removed and therefore further processing with structure-from-motion and multi-view stereo techniques becomes possible.

As said earlier, the images shown in the figures are individual sample frames taken from videos. We actually carry out the process for all frames in the video. Although there is no temporal information being taken into account when processing the video i.e. each frame is individually processed, we have seen that the resulting caustic-free video is smoothly varying and consistent across sequential frames. Figure 10 shows the saliency maps and caustic-free images are consistent

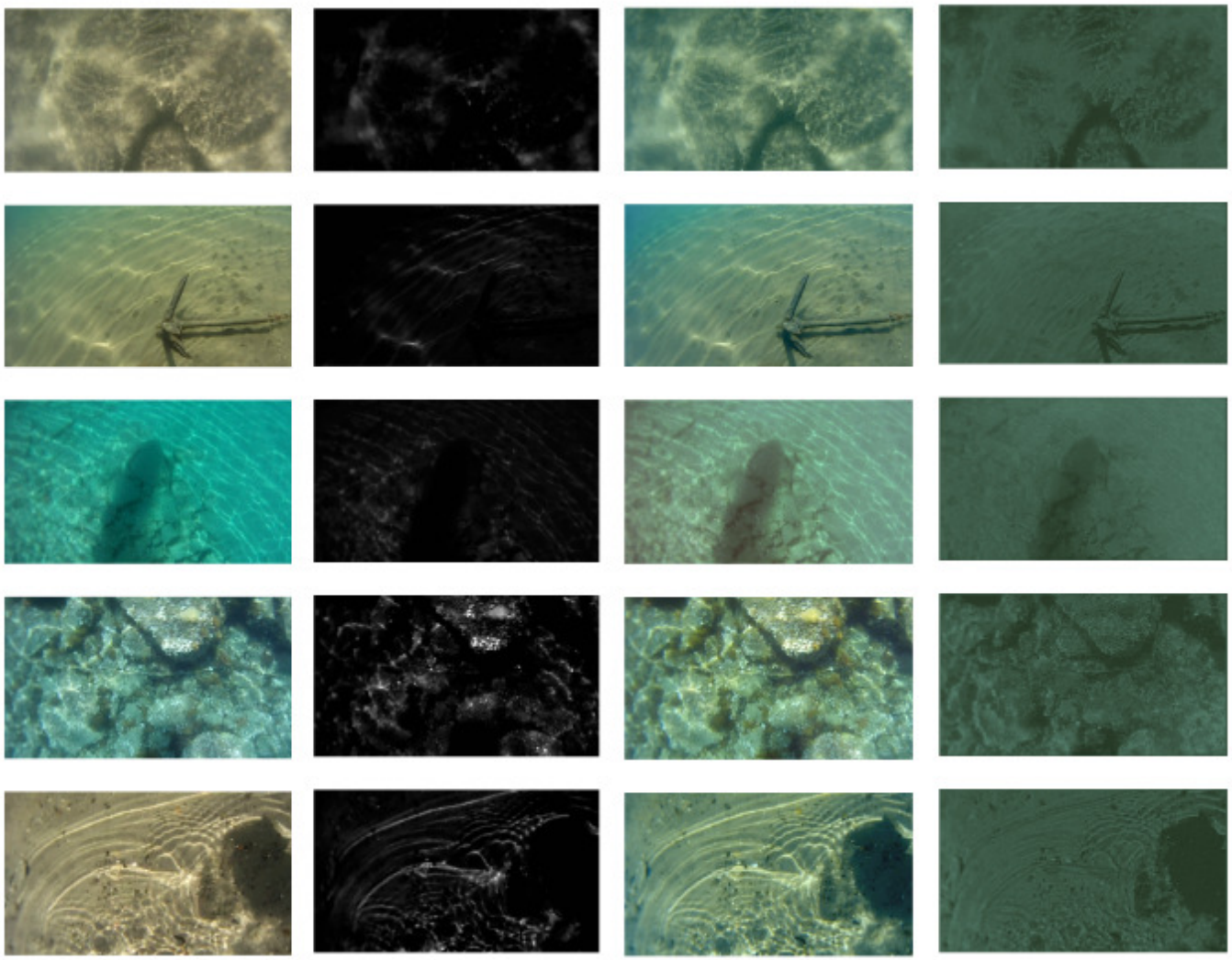
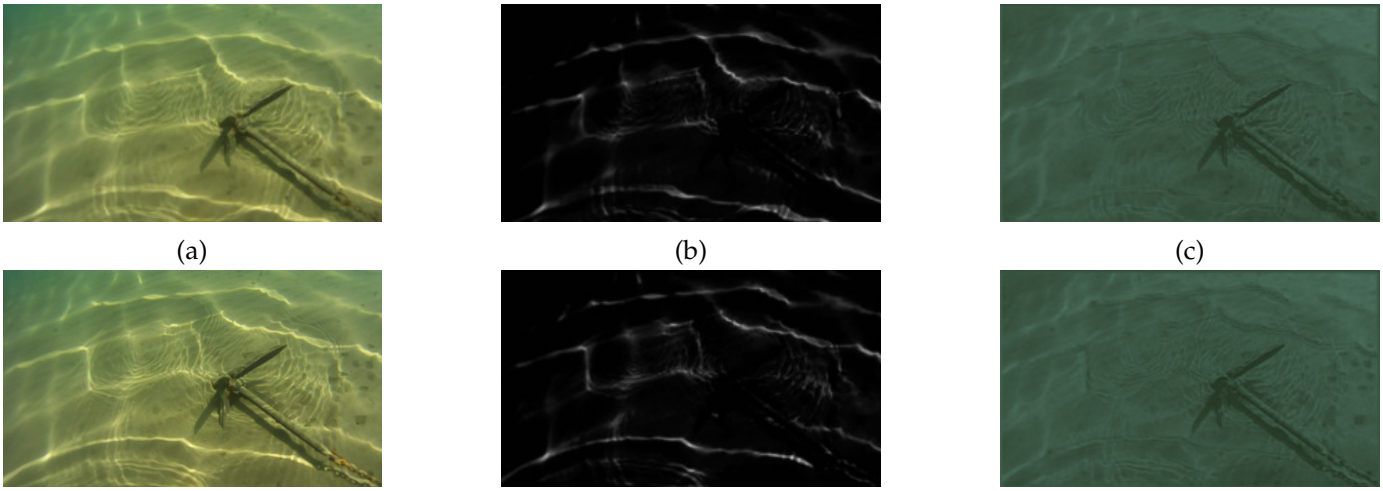


Fig. 9: First column: The original RGB images for five selected scenes which contain caustics of varying characteristics. Second column: The RGB images after color transfer and histogram matching. Third column: The results generated by DeepCaustics on the original images *without* color transfer and histogram matching i.e. first column. Fourth column: The caustic-free images generated by DeepCaustics on the color matched RGB images shown in the second column. The caustics have been successfully removed and therefore further processing with structure-from-motion and multi-view stereo techniques is possible. Please refer to supplemental videos for higher resolution visualizations.

between frames 25, 30 although there is considerable camera motion as well as caustic motion between them. Please refer to supplemental material for complete sequences of entire videos.

An extreme case of caustics produced by turbulent water motion which causes caustics of varying frequencies which are not well defined in terms of structure is shown in Figure 11. Similarly Figure 12 shows another complex example of caustics appearing on rocks with barnacles. Despite the complexity of the scene the proposed approach is able to correctly classify pixels as caustics/non-caustics and remove them. An example of this is the shadow of the boat being consistently classified as non-caustic throughout the video sequence.



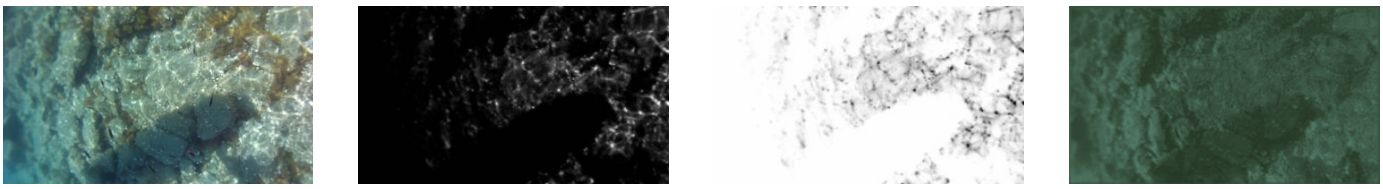
(a) (b) (c)

(d) (e) (f)

Fig. 10: Frames 25 and 30 are processed independently i.e. no temporal information is used. The saliency maps and caustic-free images are consistent although considerable motion [camera, caustics] has occurred between the two frames.

(a) (b) (c) (d)

Fig. 11: An extreme example of caustics caused by turbulent water movement. (a) The RGB image. (b) The color transferred images. (c) The saliency map generated by SalienceNet. (d) The caustic-free image generated by DeepCaustics.



(a) (b) (c) (d)

Fig. 12: Another complex example of caustics appearing on rocks barnacles. (a) The original RGB image. (b) SalienceNet's result. (c) Color-inverted saliency map. Note that objects like the shadow of the boat are consistently classified as non-caustics throughout the video sequence. (d) The caustic-free image generated by DeepCaustics. Note the occasional loss of color information due to processing.

5.2 Reconstruction

In order evaluate the results of our work we constructed a 3D object from a sequence of thresholded images and a sequence of images generated by DeepCaustics. This was done to demonstrate the effectiveness of our method relative to a simple thresholding of brightness in an image.

The 3D object consists of an block of concrete found underwater. We construct a 3D object

(a) (b) (c)

Fig. 13: Examples of the images that are used in the 3D reconstruction. (a) The original RGB image. (b) The thresholded image. (c) DeepCaustic generated image placed back into its original colorspace.

using Agisoft's PhotoScan (SfM) from our image sequences, then we extract planes from four different sides of the cube. From these planes we calculate (1) a surface fitting metric evaluating the "smoothness of our reconstruction" and (2) the orthogonality between all other faces. All faces, except opposing ones, should have an orthogonality of 90 degrees to represent how they are perpendicular. Opposing faces should be parallel. The metrics are defined below.

5.2.1 Surface fitting metric

A surface Π is reconstructed and a plane $\Pi_{fitted} = \langle \alpha, \beta, \gamma, \delta \rangle$ with normal $N_{\Pi_{fitted}} = \langle n_x, n_y, n_z \rangle$ is fitted on the resulting $\aleph$ 3D points. The plane fitting is performed using RANSAC and the average error E_{avg} and average $RMSE$ is computed as follows,

$$E_{avg} = \sum_{i=0}^{\aleph} |\delta - (\alpha \times \aleph_i^x + \beta \times \aleph_i^y + \gamma \times \aleph_i^z)|/||\aleph|| \tag{4}$$

$$RMSE = \sqrt{\frac{\sum_{i=0}^{\aleph} \{\delta - (\alpha \times \aleph_i^x + \beta \times \aleph_i^y + \gamma \times \aleph_i^z)\}^2}{||\aleph||}} \tag{5}$$

We reconstruct several different planar surfaces shown in Table 2 and perform plane fitting using RANSAC. This metric is measured as the distance of all points from the fitted surface in terms of the average error E_{avg} and the root-mean-square error $RMSE$.

5.2.2 Orthogonality metric

A set of perpendicular surfaces are reconstructed and four planes $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ are fitted respectively using RANSAC. The orthogonality metric is defined as the magnitude of the six dimensional vector containing the measured angles between the four planes in terms of the dot product as follows,

$$E_{ortho} = ||\langle dot(N^1_{\Pi_{fitted}}, N^2_{\Pi_{fitted}}), dot(N^1_{\Pi_{fitted}}, N^3_{\Pi_{fitted}}), dot(N^1_{\Pi_{fitted}}, N^4_{\Pi_{fitted}}), ..., dot(N^3_{\Pi_{fitted}}, N^4_{\Pi_{fitted}}) \rangle||_{L_2} \tag{6}$$

We reconstructed the concrete block object containing orthogonal planes. For each of the four planes a linear surface is fitted using RANSAC. This metric is measured as the angle formed between the four planes as shown in Table 3.

	Threshold	DeepCaustics
Top	0.8276	0.9690
Front	0.9965	0.9974
Left	0.9787	0.9882
Right	0.9207	0.9564

TABLE 2: Surface fitting metric for different faces of the 3D reconstructed cube.

	Top	Front	Left	Right
Top	-	90.99	91.86	85.76
Front	56.65	-	92.29	56.28
Left	56.98	57.27	-	31.75
Right	57.26	52.83	35.47	-

TABLE 3: Orthogonality between different faces of the 3D reconstructed cube. The upper triangle of the table are results from images processed with DeepCaustics. The lower triangle contains results from images where brightness has been thresholded.

5.3 Discussion

It is somewhat surprising that a small network achieved the success that it did. Although this may be due to some underlying simple properties of the synthetic data set that the network was able to exploit, the fact that good results were achieved on the real data set suggests that this is not a sufficient explanation. Alternatively, it may be that there is some simple property related to caustics, such as brightness or contrast, that is present in both the synthetic and real data set, which the network was able to pick up on, even with relatively few parameters. Small data sets, like our own, lead to over-fitting very early in the learning phase. Having a larger training set would increase the performance of our network as it learns to generalize to a wider set of inputs. Our small set of real test images resulted in biased evaluations as we could not test performance on a wider variety of caustic images. Other examples of learning from small datasets are not unheard of [24]. However, while the size of the data set affects performance our greatest liability is the quality of our dataset. Having real image pairs with and without caustics would allow the network to learn more appropriate features applicable to our test set.

6 Conclusion and Future Work

Caustics in shallow underwater videos cause significant problems downstream in further processing and analysing of these images. Earlier methods have primarily focused on image enhancement techniques to ameliorate these problems, albeit with limited success. Further to the best of our knowledge there have been no significant attempts to explicitly classify caustics in images. In this work we have reported an elegant deep-learning solution to classification and removal of caustics from shallow underwater images. Given the difficulty to obtain ground truth for the large volumes of pixel data in underwater videos, our solution uses a small synthetic dataset created using standard 3D modelling software. In creating the synthetic data we create two ground truth components, a caustics confidence mask and a caustic-free scene. We use the

first caustics confidence component to train a CNN called SalienceNet which yields a saliency value (confidence of being a caustic) per pixel. Using the saliency as the alpha value in RGBA format and the second caustic-free component as ground truth we train another CNN called DeepCaustics to yield caustic-free versions of input images with caustics. Our tests on a number of real world underwater videos show that our solution yields good results, that would certainly make further processing of these images more reliable and robust.

As can be seen from the results, even with these relatively small networks and small synthetic training dataset we were able to transfer the learning to real world data quite effectively. Investigating this further would be a challenging and interesting problem for the immediate future, in particular looking into detail at cases where this fails. Our current framework does not directly make use of the temporal information present in the way caustics change in successive frames of the scene. We would like to upgrade our method to explicitly use such temporal information and see if even better results can be obtained. We would also like to explore the possibility of extending this methodology to other complex phenomena in images which cause similar problems, such as shadows. Lastly, we would certainly like to revisit the unsupervised learning approach and experiment more systematically to see if it can be made to work.

ACKNOWLEDGMENTS

This research is based upon work supported by the Natural Sciences and Engineering Research Council of Canada Grants No. N01670 (Discovery Grant) and by the i-MareCulture project (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE, Digital Heritage) that has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 727153.

REFERENCES

[1] I. Katsouri, A. Tzanavari, K. Herakleous, and C. Poullis, "Visualizing and assessing hypotheses for marine archaeology in a vr cave environment," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 8, no. 2, p. 10, 2015.

[2] P. Drap, D. Merad, J.-M. Boï, J. Seinturier, D. Peloso, C. Reidinger, G. Vannini, M. Nucciotti, and E. Pruno, "Photogrammetry for medieval archaeology: A way to represent and analyse stratigraphy," in *Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on*. IEEE, 2012, pp. 157–164.

[3] E. Trabes and M. A. Jordan, "Self-tuning of a sunlight-deflickering filter for moving scenes underwater," in *Information Processing and Control (RPIC), 2015 XVI Workshop on*. IEEE, 2015, pp. 1–6.

[4] N. Gracias, S. Negahdaripour, L. Neumann, R. Prados, and R. Garcia, "A motion compensated filtering approach to remove sunlight flicker in shallow water images," in *OCEANS 2008*. IEEE, 2008, pp. 1–7.

[5] N. Gracias and J. Santos-Victor, "Underwater video mosaics as visual navigation maps," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, 2000.

[6] A. Shihavuddin, N. Gracias, and R. Garcia, "Online sunflicker removal using dynamic texture prediction." in *VISAPP (1)*, 2012, pp. 161–167.

[7] N. Joshi and M. F. Cohen, "Seeing mt. rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal," in *Computational Photography (ICCP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–8.

[8] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.

[9] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, p. 72, 2008.

[10] Y. Y. Schechner and N. Karpel, "Attenuating natural flicker patterns," in *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, vol. 3. IEEE, 2004, pp. 1262–1268.

[11] Y. Swirski and Y. Y. Schechner, "3deflicker from motion," in *Computational Photography (ICCP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–9.

[12] E. Trabes and M. Jordan, "On-line filtering of sunlight caustic waves in underwater scenes in motion," in *evaluation in 7th International Scientific Conference on Physics and Control*, 2015, pp. 19–22.

[13] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[14] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1434–1441.

[15] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning–ICANN 2011*. Springer, 2011, pp. 52–59.
[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.
[17] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.
[18] D. Coltuc, P. Bolon, and J.-M. Chassery, "Exact histogram specification," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1143–1152, 2006.
[19] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688
[20] H. W. Jensen, *Realistic image synthesis using photon mapping*. Ak Peters Natick, 2001, vol. 364.
[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
[23] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," *arXiv preprint arXiv:1511.08861*, 2015.
[24] P. Ren, Y. Dong, S. Lin, X. Tong, and B. Guo, "Image based relighting using neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 111, 2015.

Timothy Forbes is currently pursuing his Masters in Computer Science at Concordia University. When he completed his Bachelor of Computer Science in 2016 at the same university he resisted his urge to move to a new country, and stayed so he could pursue his interest in research. His urge to move stems from an international childhood; he was born in Ecuador and then moved to Peru, Sweden, France, China, and finally Montreal, Canada.

Mark Goldsmith received his Ph.D. from Concordia University in 2015. His research interests include neural networks, pseudorandom number generation, dynamical systems, and the prediction of epileptic seizures.

Sudhir Mudur obtained his Bachelor of Technology (honours) in 1970 from IIT Bombay and his PhD in 1976 from the Tata Institute of Fundamental Research in Mumbai, India. His interest in computer graphics started with his undergraduate thesis project. Since then he has been actively researching the field of computer graphics, particularly the areas 3D modelling, global illumination, virtual environments and applications in CAD/CAM and entertainment. Over this period of more than 4 decades, he has published papers in top computer graphics venues and supervised a large number of doctoral and graduate students, many of whom are well established in the field. His work in the areas of robust geometric computing using interval arithmetic and in global illumination models is well cited. Mudur is currently a professor and chair of the department of computer science and software engineering at Concordia university. Mudur is a senior member of IEEE, member of ACM, member of Eurographics and a SIGGRAPH Pioneer Group member.

Charalambos Poullis was born in Nicosia, Cyprus, in 1978. He received the B.Sc. degree in Computing and Information Systems with First Class Honors from the University of Manchester, UK, in 2001, and the M.Sc. in Computing Science with specialisation in Multimedia and Creative Technologies, and Ph.D. in Computer Science from the University of Southern California (USC), Los Angeles, USA, in 2003 and 2008, respectively. Since August 2015, he has been with the Department of Computer Science and Software Engineering at the Faculty of Engineering and Computer Science at Concordia University where he also serves as the Director of the Immersive and Creative Technologies (ICT) lab. His current research interests lie at the intersection of computer vision and computer graphics. More specifically, he is involved in fundamental and applied research covering the following areas: acquisition technologies & 3D reconstruction, photo-realistic rendering, feature extraction & classification, virtual & augmented reality. Charalambos is a member of the Association for Computing Machinery(ACM); Institute of Electrical and Electronics Engineers (IEEE) Computer Society; Marie Curie Alumni Association (MCAA); ACM Cyprus Chapter, where he also served in the management committee between 2010-2015; and British Machine Vision Association (BMVA). Charalambos has been serving as a regular reviewer in numerous premier conferences and journals since 2003.